



注意：この日本語版文書は参考資料としてご利用ください。最新情報は必ずオリジナルの英語版をご参照願います。

PIC16(L)F1764/5/8/9

14/20 ピン、8 ビットフラッシュ マイクロコントローラ

説明

PIC16(L)F1764/5/8/9 はインテリジェント アナログ モジュールとデジタル モジュールを内蔵し、幅広い種類の機能と最終製品に最適です。これらの 14/20 ピン デバイスは 10 ビット ADC、オペアンプ、ゼロクロス検出、大電流 I/O、通信、ペリフェラル ピンセレクト等便利なモジュールを多数内蔵しており、設計の柔軟性を必要とするアプリケーションに最適です。

主な特長

- C コンパイラ向けに最適化された RISC アーキテクチャ
- わずか 49 個の命令
- 動作速度：
 - DC ~ 32 MHz のクロック入力
 - 125 ns の最小命令サイクル
- 割り込み機能
- 16 段のハードウェア スタック
- 最大 4 個の 8 ビットタイマ
- 最大 3 個の 16 ビットタイマ
- パワーオン リセット (POR)
- 設定可能なパワーアップ タイマ (PWRT)
- トリップポイントを選択可能なブラウンアウト リセット (BOR)
- 拡張ウォッチドッグ タイマ (EWDT):
 - 低消費電力の 31 kHz WDT
 - ソフトウェアで選択可能なプリスケアラ
 - ソフトウェアで ON/OFF 可

メモリ

- 最大 14 KB のフラッシュ プログラムメモリ
- 最大 1024 バイトのデータ RAM
- 直接、間接、相対アドレッシング モード
- 高書き込み耐性フラッシュ (HEF):
 - 128 バイトの不揮発性データストレージ
 - 消去 / 書き込みサイクル: 10 万回

動作特性

- 動作電圧レンジ：
 - 1.8 ~ 3.6 V (PIC16LF1764/5/8/9)
 - 2.3 ~ 5.5 V (PIC16F1764/5/8/9)
- 温度レンジ：
 - 産業用温度レンジ: -40 ~ 85 °C
 - 拡張温度レンジ: -40 ~ 125 °C

超低消費電力 (XLP) 機能

- スリープ: 1.8 V で 50 nA (typ.)
- ウォッチドッグ タイマ: 1.8 V で 500 nA (typ.)
- セカンダリ オシレータ: 32 kHz で 500 nA
- 動作時電流：
 - 1.8 V、32 kHz で 8 µA (typ.)
 - 1.8 V で 32 µA/MHz (typ.)
- 低消費電力 BOR (LPBOR):
 - 200 nA (スリープ中)

デジタル周辺モジュール

- 構成可能なロジックセル (CLC):
 - 最大 4 つの入力を選択可能な最大 3 個の CLC
 - 組み合わせロジックと順序ロジックを内蔵
- 最大 2 個の相補出力ジェネレータ (COG):
 - プッシュプル、フルブリッジ、ステアリング モード
- 最大 2 個のキャプチャ / コンペア / PWM (CCP) モジュール
- パルス幅変調器 (PWM):
 - 最大 2 個の 10 ビット PWM
 - 最大 2 個の 16 ビット PWM
- ペリフェラル ピンセレクト (PPS):
 - 任意のデジタルピンを出力に設定
- シリアル通信：
 - 拡張 USART (EUSART)
 - SPI、I²C™、RS-232、RS-485、LIN 互換
 - 自動 baud レート検出、起動時に自動復帰
- 最大 18 本の I/O ピン：
 - 個別にプログラム可能なプルアップ
 - スルーレート制御
 - エッジ選択可能な状態変化割り込み
- 最大 2 個のデータ信号モジュレータ (DSM)

インテリジェント アナログ周辺モジュール

- 10 ビット A/D コンバータ (ADC):
 - 最大 12 本の外部チャンネル
 - スリープ中でも変換可能
- 最大 2 個のオペアンプ (OPA):
 - 内部チャンネルと外部チャンネルを選択可能
- 最大 4 個の高速コンパレータ (COMP):
 - 最大 5 個の外部反転入力
 - 最大 8 個の外部非反転入力
 - 固定参照電圧を非反転入力に接続可能
 - 外部からコンパレータ出力にアクセス可能
- D/A コンバータ (DAC):
 - 最大 2 個の 10 ビット DAC
 - 最大 2 個の 5 ビット DAC

PIC16(L)F1764/5/8/9

インテリジェント アナログ周辺モジュール (続き)

- 参照電圧 :
 - 固定参照電圧 (FVR): 出力レベル 1.024 V、2.048 V、4.096 V
- ゼロクロス検出器 (ZCD):
 - 高電圧 AC 信号を検出
- プログラマブル ランプ ジェネレータ (PRG):
 - スロープ補償
 - ランプ波生成
- 大電流駆動 I/O:
 - 電流容量 : 5 V 時に最大 100 mA

クロック構造

- 16 MHz の内部オシレータ :
 - 工場出荷時に ±1% に校正済み
 - 周波数は 32 MHz ~ 31 kHz から選択可能
- 31 kHz 低消費電力内部オシレータ
- 4x 位相ロックループ (PLL):
 - 最大 32 MHz の内部動作が可能
- 外部オシレータ ブロック :
 - 最大 32 MHz の 3 つの外部クロックモード

表 1: PIC16(L)F1764/5/8/9 ファミリのデバイス

デバイス	データシート インデックス	プログラマブルフラッシュ (ワード /KB)	書き換え耐性フラッシュ (バイト)	データ SRAM (バイト)	I/O ピン ⁽²⁾	8/16 ビットタイマ	コンパレータ	10 ビット ADC (ch)	5/10 ビット DAC	GCP	10/16 ビット PWM	COG	データ信号モジュレータ	CLC	オペアンプ	ゼロクロス検出器	プログラマブル ランプジェネレータ	大電流 I/O	ペリフェラルピンセレクト	EUSART	I ² C™/SPI	デバッグ ⁽¹⁾
PIC16(L)F1764	(A)	4096/7	128	512	12	4/3	2	8	1/1	1	1/1	1	1	3	1	1	1	2	あり	1	1	I/H
PIC16(L)F1765	(A)	8192/14	128	1024	12	4/3	2	8	1/1	1	1/1	1	1	3	1	1	1	2	あり	1	1	I/H
PIC16(L)F1768	(A)	4096/7	128	512	18	4/3	4	12	2/2	2	2/2	2	2	3	2	1	2	2	あり	1	1	I/H
PIC16(L)F1769	(A)	8192/14	128	1024	18	4/3	4	12	2/2	2	2/2	2	2	3	2	1	2	2	あり	1	1	I/H

Note 1: デバッグ方法 : (I) – 内蔵、(H) – ICD ヘッダ使用、(E) – エミュレーション製品
2: 1 本は入力専用です。

データシート インデックス : (本書では、網掛けしていないデバイスについて説明します。)

A. DS-40001775 PIC16(L)F1764/5/8/9 データシート、14/20 ピン、8 ビットフラッシュ マイクロコントローラ

Note: その他の小型パッケージの在庫 / 供給状況とマーキング情報については、
<http://www.microchip.com/packaging> を参照するか、販売代理店までお問い合わせください。

表 2: パッケージ

パッケージ	PDIP	SOIC	TSSOP	QFN	SSOP
PIC16(L)F1764	•	•	•	•	
PIC16(L)F1765	•	•	•	•	
PIC16(L)F1768	•	•		•	•
PIC16(L)F1769	•	•		•	•

Note: ピンの詳細は変更される場合があります。

ピン配置図

図 1: 14 ピン PDIP/SOIC/TSSOP

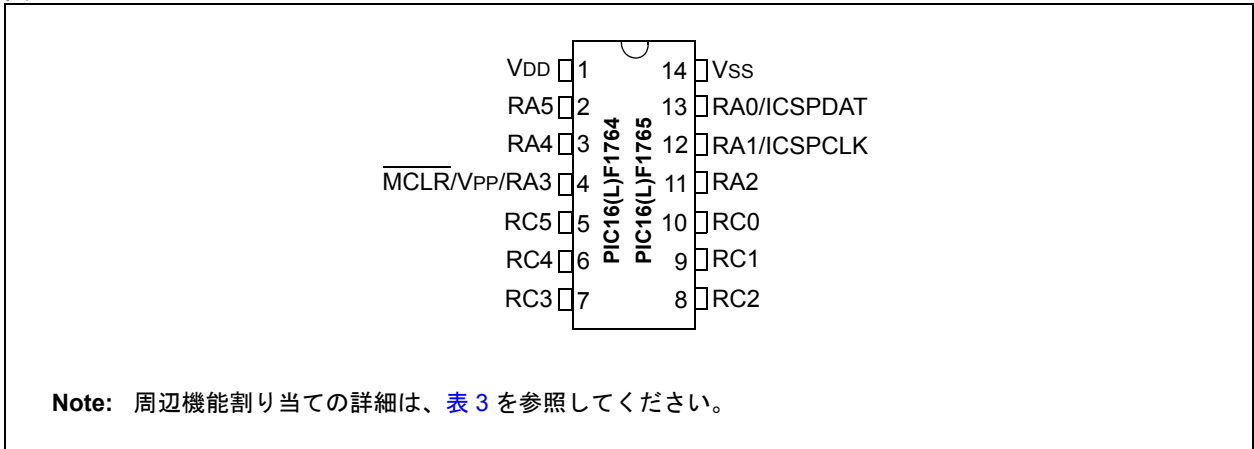
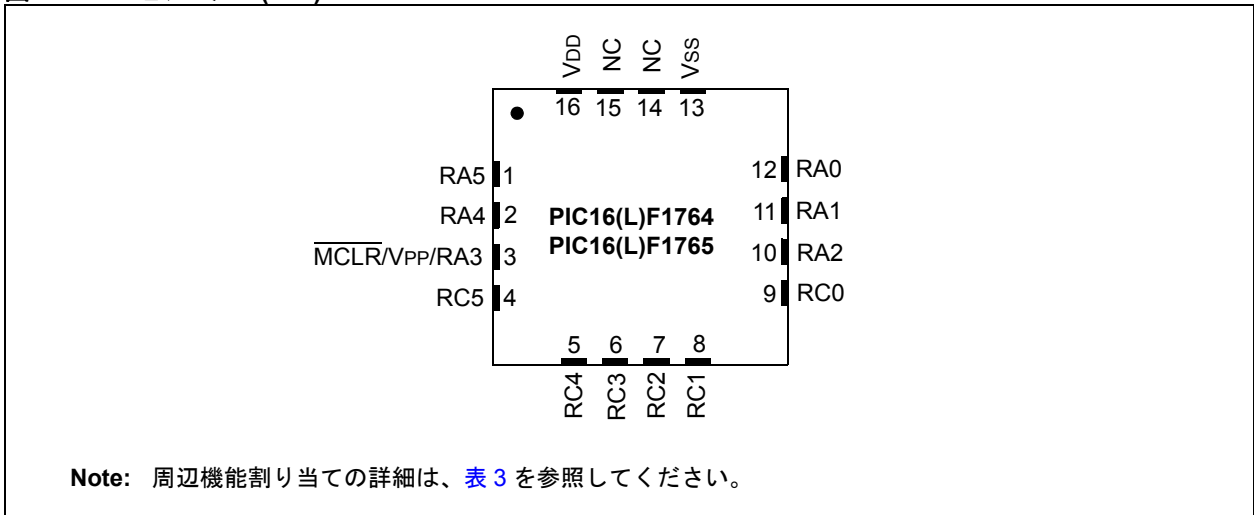


図 2: 16 ピン QFN (4x4)



PIC16(L)F1764/5/8/9

図 3: 20 ピン PDIP/SOIC/SSOP

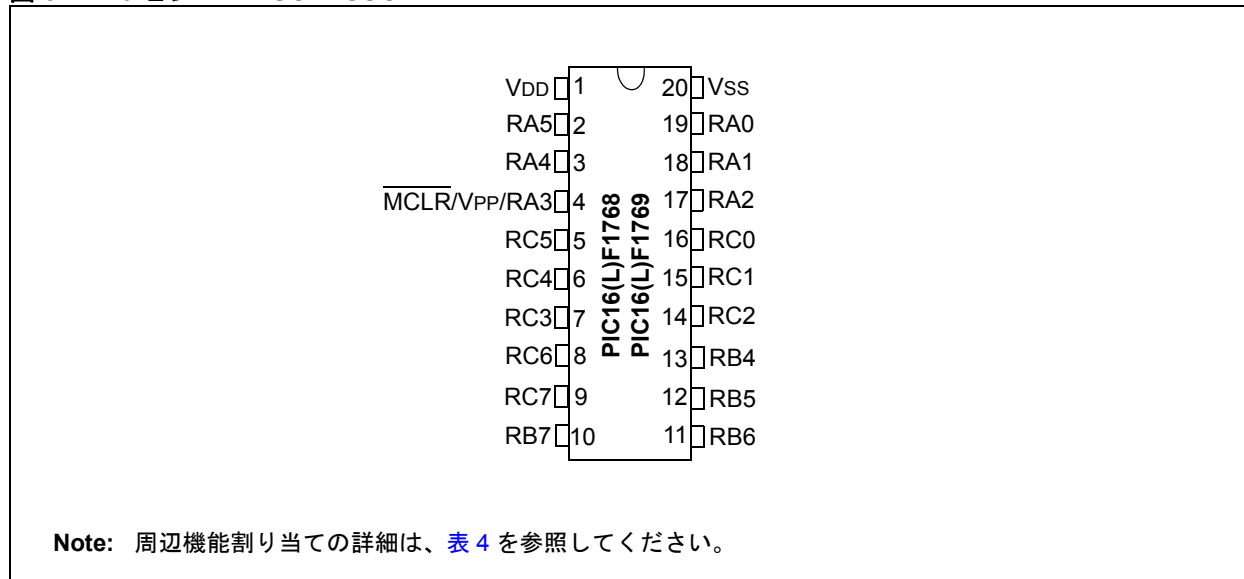
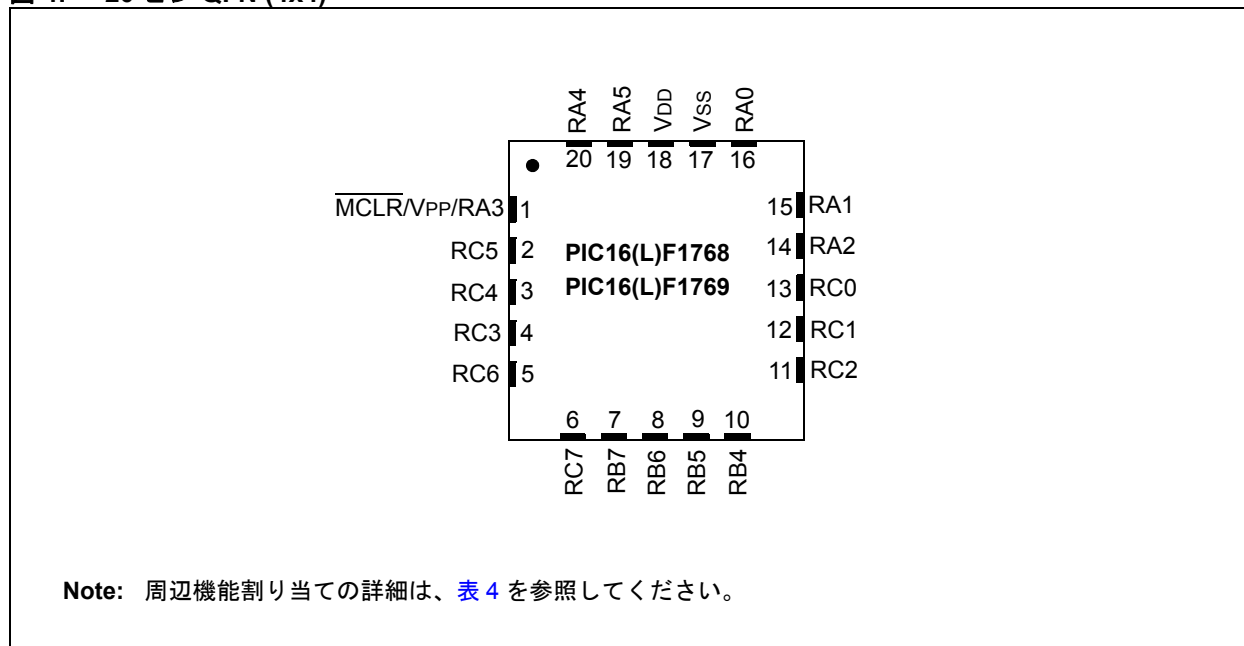


図 4: 20 ピン QFN (4x4)



ピン割り当て表

表 3: 14/16 ピンデバイスのピン割り当て (PIC16(L)F1764/5)

I/O	14ピン PDIP/SOIC/TSSOP	16ピン QFN	ADC	参照電圧	DAC	オペアンプ	コンパレータ	ゼロクロス	プログラマブル ランプジェネレータ	タイマ	PWM	CCP	COG	CLC	モジュレータ	EUSART	MSSP	割り込み	プルアップ	大電流	基本
RA0	13	12	AN0	VREF- DAC1REF- DAC3REF-	DAC1OUT1 DAC3OUT1	—	C1IN0+	—	—	—	—	—	—	—	—	—	—	IOC	あり	—	ICSPDAT
RA1	12	11	AN1	VREF+ DAC1REF+ DAC3REF+	—	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	—	—	IOC	あり	—	ICSPCLK
RA2	11	10	AN2	—	—	—	—	ZCD	—	T0CKI ⁽¹⁾	—	—	COG1IN ⁽¹⁾	—	—	—	—	INT ⁽¹⁾ IOC	あり	—	—
RA3	4	3	—	—	—	—	—	—	—	T6CKI ⁽¹⁾	—	—	—	—	MD1CH ⁽¹⁾	—	—	IOC	あり	—	V _{PP} MCLR
RA4	3	2	AN3	—	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	MD1CL ⁽¹⁾	—	—	IOC	あり	—	OSC2 CLKOUT
RA5	2	1	—	—	—	—	—	—	—	T1CKI ⁽¹⁾ T2CKI ⁽¹⁾ SOSCI	—	—	—	CLCIN3 ⁽¹⁾	MD1MOD ⁽¹⁾	—	—	IOC	あり	—	OSC1 CLKIN
RC0	10	9	AN4	—	—	OPA1IN+	C2IN0+	—	—	T5CKI ⁽¹⁾	—	—	—	—	—	—	SCL ⁽¹⁾ SCK ^(1,3)	IOC	あり	—	—
RC1	9	8	AN5	—	—	OPA1IN-	C1IN1- C2IN1-	—	—	T4CKI ⁽¹⁾	—	—	—	CLCIN2 ⁽¹⁾	—	—	SDI ⁽¹⁾ SDA ^(1,3)	IOC	あり	—	—
RC2	8	7	AN6	—	—	OPA1OUT	C1IN2- C2IN2-	—	PRG1IN0	—	—	—	—	—	—	—	—	IOC	あり	—	—
RC3	7	6	AN7	—	—	—	C1IN3- C2IN3-	—	—	T5G ⁽¹⁾	—	—	—	CLCIN0 ⁽¹⁾	—	—	SS ⁽¹⁾	IOC	あり	—	—
RC4	6	5	—	—	—	—	—	—	PRG1R ⁽¹⁾	T3G ⁽¹⁾	—	—	—	CLCIN1 ⁽¹⁾	—	CK ⁽¹⁾	—	IOC	あり	あり	—
RC5	5	4	—	—	—	—	—	—	PRG1F ⁽¹⁾	T3CKI ⁽¹⁾	—	CCP1 ⁽¹⁾	—	—	—	RX ^(1,3)	—	IOC	あり	あり	—
V _{DD}	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	V _{DD}
V _{SS}	14	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	V _{SS}
OUT ⁽²⁾	—	—	—	—	—	—	C1OUT	—	—	—	PWM3	CCP1	COG1A	CLC1OUT	MD1OUT	DT ⁽³⁾	SDO	INT	—	—	—
	—	—	—	—	—	—	C2OUT	—	—	—	PWM5	—	COG1B	CLC2OUT	—	TX	SDA ⁽³⁾	—	—	—	—
	—	—	—	—	—	—	—	—	—	—	—	—	COG1C	CLC3OUT	—	CK	SCK	—	—	—	—
	—	—	—	—	—	—	—	—	—	—	—	—	COG1D	—	—	—	SCL ⁽³⁾	—	—	—	—

- Note**
- 1: 周辺モジュールの既定値入力です。入力はPPS入力選択レジスタで他の任意のピンに移動できます。
 - 2: 全てのピン出力の既定値は、PORTラッチデータです。PPS出力選択レジスタで、任意のピンをデジタル周辺モジュール出力として選択できます。
 - 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

表 4: 20 ピンデバイスのピン割り当て (PIC16(L)F1768/9)

I/O	20 ピン PDIP/SSOP	20 ピン QFN	ADC	参照電圧	DAC	オペアンプ	コンパレータ	ゼロクロス	プログラマブル ランプジェネレータ	タイマ	PWM	CCP	COG	CLC	モジュールタ	EUSART	MSSP	割り込み	プルアップ	大電流	基本
RA0	19	16	AN0	VREF- DAC1REF- DAC2REF- DAC3REF- DAC4REF-	DAC1OUT1 DAC2OUT1 DAC3OUT1 DAC4OUT1	—	C1IN0+ C3IN0+	—	—	—	—	—	—	—	—	—	—	IOC	あり	—	ICSPDAT
RA1	18	15	AN1	VREF+ DAC1REF+ DAC2REF+ DAC3REF+ DAC4REF+	—	—	C1IN0- C2IN0- C3IN0- C4IN0-	—	—	—	—	—	—	—	—	—	—	IOC	あり	—	ICSPCLK
RA2	17	14	AN2	—	—	—	—	ZCD	—	T0CKI ⁽¹⁾	—	—	COG1IN ⁽¹⁾ COG2IN ⁽¹⁾	—	—	—	—	INT ⁽¹⁾ IOC	あり	—	—
RA3	4	1	—	—	—	—	—	—	—	T6CKI ⁽¹⁾	—	—	—	—	MD1CH ⁽¹⁾ MD2CH ⁽¹⁾	—	—	IOC	あり	—	V _{PP} MCLR ICD
RA4	3	20	AN3	—	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	MD1CL ⁽¹⁾ MD2CL ⁽¹⁾	—	—	IOC	あり	—	OSC2 CLKOUT
RA5	2	19	—	—	—	—	—	—	—	T1CKI ⁽¹⁾ T2CKI ⁽¹⁾ SOSCI	—	—	—	CLCIN3 ⁽¹⁾	MD1MOD ⁽¹⁾ MD2MOD ⁽¹⁾	—	—	IOC	あり	—	OSC1 CLKIN
RB4	13	10	AN10	—	—	OPA1IN0-	—	—	—	—	—	—	—	—	—	—	SDI ⁽¹⁾ SDA ^(1,3)	IOC	あり	—	—
RB5	12	9	AN11	—	—	OPA1IN0+	—	—	—	—	—	—	—	—	—	RX ^(1,3)	—	IOC	あり	—	—
RB6	11	8	—	—	—	—	C1IN1+ C3IN1+	—	—	—	—	—	—	—	—	—	SCL ⁽¹⁾ SCK ^(1,3)	IOC	あり	—	—
RB7	10	7	—	—	—	—	C2IN1+ C4IN1+	—	—	—	—	—	—	—	—	CK ⁽¹⁾	—	IOC	あり	—	—
RC0	16	13	AN4	—	—	—	C2IN0+ C4IN0+	—	—	T5CKI ⁽¹⁾	—	—	—	—	—	—	—	IOC	あり	—	—
RC1	15	12	AN5	—	—	—	C1IN1- C2IN1- C3IN1- C4IN1-	—	—	T4CKI ⁽¹⁾	—	—	—	CLCIN2 ⁽¹⁾	—	—	—	IOC	あり	—	—
RC2	14	11	AN6	—	—	OPA1OUT OPA2IN1- OPA2IN1+	C1IN2- C2IN2-	—	PRG1IN0 PRG2IN1	—	—	—	—	—	—	—	—	IOC	あり	—	—

Note 1: 周辺モジュールの既定値入力です。入力は PPS 入力選択レジスタで他の任意のピンに移動できます。
 2: 全てのピン出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタで、任意のピンをデジタル周辺モジュール出力として選択できます。
 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

表 4: 20 ピンデバイスのピン割り当て (PIC16(L)F1768/9) (続き)

I/O	20ピン PDIP/SOIC/SSOP		ADC	参照電圧	DAC	オペアンプ	コンパレータ	ゼロクロス	プログラマブル ランプジェネレータ	タイマ	PWM	CCP	COG	CLC	モジュレータ	EUSART	MSSP	割り込み	プルアップ	大電流	基本	
	7	4																				
RC3	7	4	AN7	—	—	OPA2OUT OPA1IN1- OPA1IN1+	C1IN3- C2IN3- C3IN3- C4IN3-	—	PRG2IN0 PRG1IN1	T5G ⁽¹⁾	—	CCP2 ⁽¹⁾	—	CLCIN0 ⁽¹⁾	—	—	—	IOC	あり	—	—	
RC4	6	3	—	—	—	—	—	—	PRG1R ⁽¹⁾ PRG2R ⁽¹⁾	T3G ⁽¹⁾	—	—	—	CLCIN1 ⁽¹⁾	—	—	—	IOC	あり	あり	—	
RC5	5	2	—	—	—	—	—	—	PRG1F ⁽¹⁾ PRG2F ⁽¹⁾	T3CKI ⁽¹⁾	—	CCP1 ⁽¹⁾	—	—	—	—	—	IOC	あり	あり	—	
RC6	8	5	AN8	—	—	OPA2IN0-	—	—	—	—	—	—	—	—	—	—	SS ⁽¹⁾	IOC	あり	—	—	
RC7	9	6	AN9	—	—	OPA2IN0+	—	—	—	—	—	—	—	—	—	—	—	IOC	あり	—	—	
V _{DD}	1	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
V _{SS}	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
OUT ⁽²⁾	—	—	—	—	—	—	C1OUT	—	—	—	PWM3	CCP1	COG1A	CLC1OUT	MD1OUT	DT ⁽³⁾	SDO	—	—	—	—	
	—	—	—	—	—	—	C2OUT	—	—	—	PWM4	CCP2	COG1B	CLC2OUT	MD2OUT	TX	SDA ⁽³⁾	—	—	—	—	
	—	—	—	—	—	—	C3OUT	—	—	—	PWM5	—	COG1C	CLC3OUT	—	CK	SCK	—	—	—	—	
	—	—	—	—	—	—	C4OUT	—	—	—	PWM6	—	COG1D	—	—	—	SCL ⁽³⁾	—	—	—	—	
	—	—	—	—	—	—	—	—	—	—	—	—	COG2A	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	—	—	—	—	COG2B	—	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—	—	—	—	—	COG2C	—	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—	—	—	—	—	COG2D	—	—	—	—	—	—	—	—	—

Note 1: 周辺モジュールの既定値入力です。入力は PPS 入力選択レジスタで他の任意のピンに移動できます。
2: 全てのピン出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタで、任意のピンをデジタル周辺モジュール出力として選択できます。
3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

PIC16(L)F1764/5/8/9

目次

1.0	デバイス概要	10
2.0	エンハンスド ミッドレンジ CPU	23
3.0	メモリの構成	25
4.0	デバイス コンフィグレーション	64
5.0	オシレータ モジュール (フェイルセーフ クロックモニタ機能付き)	71
6.0	リセット	89
7.0	割り込み	97
8.0	パワーダウン モード (スリープ)	110
9.0	ウォッチドッグ タイマ (WDT)	114
10.0	フラッシュ プログラムメモリ制御	119
11.0	I/O ポート	136
12.0	ペリフェラル ビンセレクト (PPS) モジュール	156
13.0	状態変化割り込み	164
14.0	固定参照電圧 (FVR)	171
15.0	温度インジケータ モジュール	174
16.0	A/D コンバータ (ADC) モジュール	176
17.0	5 ビット D/A コンバータ (DAC) モジュール	190
18.0	10 ビット D/A コンバータ (DAC) モジュール	195
19.0	コンパレータ モジュール	202
20.0	ゼロクロス検出 (ZCD) モジュール	212
21.0	Timer0 モジュール	217
22.0	Timer1/3/5 モジュール (ゲート制御対応)	220
23.0	Timer2/4/6 モジュール	231
24.0	キャプチャ / コンペア / PWM モジュール	251
25.0	10 ビットパルス幅変調 (PWM) モジュール	265
26.0	16 ビットパルス幅変調 (PWM) モジュール	271
27.0	相補出力ジェネレータ (COG) モジュール	296
28.0	構成可能なロジックセル (CLC)	331
29.0	オペアンプ (OPA) モジュール	346
30.0	プログラマブル ランプ ジェネレータ (PRG) モジュール	352
31.0	データ信号モジュレータ (DSM)	365
32.0	マスタ同期シリアルポート (MSSP) モジュール	375
33.0	拡張 USART (EUSART: Enhanced Universal Synchronous Asynchronous Receiver Transmitter)	428
34.0	インサーキット シリアル プログラミング™ (ICSP™)	459
35.0	命令セットのまとめ	461
36.0	電氣的仕様	475
37.0	DC および AC 特性の図表	509
38.0	開発サポート	522
39.0	パッケージ情報	526
補遺 A:	データシート改訂履歴	545
	Microchip 社ウェブサイト	546
	お客様向け変更通知サービス	546
	カスタマサポート	546
	製品識別システム	547

大切なお客様へ

Microchip 社は、大切なお客様に Microchip 社製品を適切にご使用頂くために、最高品質の文書を提供する事を心掛けています。このため弊社は新刊および既刊改訂版の発行を通して文書の充実と改善に継続的に取り組んで参ります。

本書に関してご質問またはご意見がございましたら、マーケティング コミュニケーション部宛てにメールでご連絡ください。メールの宛先は docerrors@microchip.com です。お客様からのご意見とご感想をお待ちしております。

最新のデータシート

本書の最新版を入手するには、弊社ウェブサイトにご登録ください。

<http://www.microchip.com>

データシートのリビジョンは、各ページの欄外下隅に記載されている文書番号で確認できます。文書番号の最後の文字がリビジョン番号を表します (例: DS30000000A_JP であれば文書 DS30000000_JP のリビジョン A)。

エラッタ

現行のデバイスに対して、データシートとの動作上の微妙な相違点と推奨回避策を説明したエラッタシートが発行される場合があります。弊社はデバイスや文書に関する問題を認識した時点でエラッタを発行します。エラッタには該当するシリコンと文書のリビジョンを明記しています。

ご使用のデバイス向けにエラッタシートが発行されているかどうかは、以下で確認できます。

• Microchip 社のウェブサイト: <http://www.microchip.com>

• 最寄りの Microchip 社営業所 (本書の巻末に記載)

お問い合わせの際は、お使いのデバイス、シリコンとデータシートのリビジョン (文書番号含む) をお知らせください。

お客様向け通知システム

弊社ウェブサイト (www.microchip.com) でご登録頂くと、弊社の全製品に関する最新情報をお受け取り頂けます。

PIC16(L)F1764/5/8/9

1.0 デバイス概要

本データシートでは、PIC16(L)F1764/5/8/9 について説明します。提供中のパッケージ構成は表 2 を参照してください。

図 1-1 に PIC16(L)F1764/5 のブロック図を示します。

図 1-2 に PIC16(L)F1768/9 のブロック図を示します。

表 1-2 と表 1-3 は、ピンの説明です。

各デバイスで利用できる周辺モジュールについては表 1-1 を参照してください。

表 1-1: デバイスの周辺モジュールのまとめ

周辺モジュール	PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
A/D コンバータ (ADC)	•	•	•	•
固定参照電圧 (FVR)	•	•	•	•
ゼロクロス検出 (ZCD)	•	•	•	•
温度インジケータ	•	•	•	•
相補出力ジェネレータ (COG)				
	COG1	•	•	•
	COG2		•	•
プログラマブル ランプ ジェネレータ (PRG)				
	PRG1	•	•	•
	PRG2		•	•
10 ビット D/A コンバータ (DAC)				
	DAC1	•	•	•
	DAC2		•	•
5 ビット D/A コンバータ (DAC)				
	DAC3	•	•	•
	DAC4		•	•
キャプチャ/コンペア/PWM (CCP/ECCP) モジュール				
	CCP1	•	•	•
	CCP2		•	•
コンパレータ				
	C1	•	•	•
	C2	•	•	•
	C3		•	•
	C4		•	•
構成可能なロジックセル (CLC)				
	CLC1	•	•	•
	CLC2	•	•	•
	CLC3	•	•	•
データ信号モジュレータ (DSM)				
	DSM1	•	•	•
	DSM2		•	•

表 1-1: デバイスの周辺モジュールのまとめ

周辺モジュール	PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter)				
	EUSART	•	•	•
マスタ同期シリアルポート				
	MSSP	•	•	•
オペアンプ				
	オペアンプ 1	•	•	•
	オペアンプ 2		•	•
10 ビットパルス幅変調器 (PWM)				
	PWM3	•	•	•
	PWM4		•	•
16 ビットパルス幅変調器 (PWM)				
	PWM5	•	•	•
	PWM6		•	•
8 ビットタイマ				
	Timer0	•	•	•
	Timer2	•	•	•
	Timer4	•	•	•
	Timer6	•	•	•
16 ビットタイマ				
	Timer1	•	•	•
	Timer3	•	•	•
	Timer5	•	•	•

1.1 レジスタおよびビット命名法

1.1.1 レジスタ名

デバイス内の同種の周辺モジュールに複数のインスタンスが存在する場合、周辺モジュール識別子、周辺モジュール インスタンス、制御識別子を連結したものを周辺モジュール制御レジスタ名とします。制御レジスタの説明では、周辺モジュール インスタンス番号を「x」で置き換えて説明します。モジュールのインスタンスが1つしかないデバイスの場合でもこの命名法を適用します。ファミリー内の他のデバイスとの互換性を維持するためです。

1.1.2 ビット名

ビット名には以下の2種類があります。

- 短縮名：ビット機能の略号
- 完全名：周辺モジュールの略号 + 短縮名

1.1.2.1 短縮ビット名

短縮ビット名はビット機能の略号の一種です。例えば、周辺モジュールの中には EN ビットを使って有効にするものがあります。この時レジスタで示されるビット名は短縮名を元にした変異形です。

短縮ビット名はCプログラムでビットを参照する際に便利です。短縮名によるビット参照の一般的なフォーマットは「**レジスタ名 bits. 短縮名**」です。例えば、COG1CON0 レジスタのイネーブルビットである EN ビットは、Cプログラムでは以下の命令でセットできます。

```
COG1CON0bits.EN = 1
```

アセンブリ プログラムでは、異なる周辺モジュールが異なるビット位置で同じ名前を使うことがあるため、一般に短縮名は適していません。インクルード ファイル生成中このような事態が生じた場合でも命名が競合しないように、同じ短縮名の全てのインスタンスにアンダースコアとそのビットが存在するレジスタ名が付加されます。

1.1.2.2 完全ビット名

完全ビット名は短縮名に周辺モジュールの略号を接頭辞として付加する事で構成します。この接頭辞は周辺モジュールで一意的のため、各完全ビット名は一意的になります。COG1 イネーブルビットの完全ビット名は、COG1 の接頭辞である G1 にイネーブルビットの短縮名 EN を添えて、G1EN という一意的ビット名になります。

完全ビット名はCとアセンブリの両方のプログラムに適しています。例えば、C では COG1CON0 イネーブルビットは以下の命令でセットできます。

```
G1EN = 1 アセンブリではこのビットは以下の命令でセットできます。
```

```
BSF COG1CON0,G1EN
```

1.1.2.3 ビットフィールド

ビットフィールドとは同じレジスタの隣り合う複数のビットを指します。ビットフィールドは短縮ビットの命名法にのみ従います。例えば、COG1CON0 レジスタの下位3ビットはモード制御ビットを含みます。このフィールドの短縮名は MD です。完全ビット名はありません。ビットフィールドはCプログラム内でのみ参照できます。以下の例では COG1 をプッシュプルモードに設定するCプログラム命令を示します。

```
COG1CON0bits.MD = 0x5;
```

ビットフィールドの個々のビットは、完全ビット名と短縮ビット名の両方で参照できます。各ビット名は、フィールド名にフィールド内のビット位置の番号を追加したものです。例えば、モードビットの最上位の短縮ビット名は MD2 であり、完全ビット名は G1MD2 です。以下の2つの例では COG1 をプッシュプルモードに設定するアセンブリ プログラム シーケンスを示します。

例 1:

```
MOVLW ~(1<<G1MD1)
ANDWF COG1CON0,F
MOVLW 1<<G1MD2 | 1<<G1MD0
IORWF COG1CON0,F
```

例 2:

```
BSF COG1CON0,G1MD2
BCF COG1CON0,G1MD1
BSF COG1CON0,G1MD0
```

1.1.3 レジスタおよびビット命名の例外

1.1.3.1 ステータスビット、割り込みビット、ミラービット

ステータス、割り込みイネーブル、割り込みフラグ、ミラーの各ビットは対象が複数の周辺モジュールにわたるレジスタに含まれています。この場合、ビット名は一意的のため接頭辞または短縮名はありません。

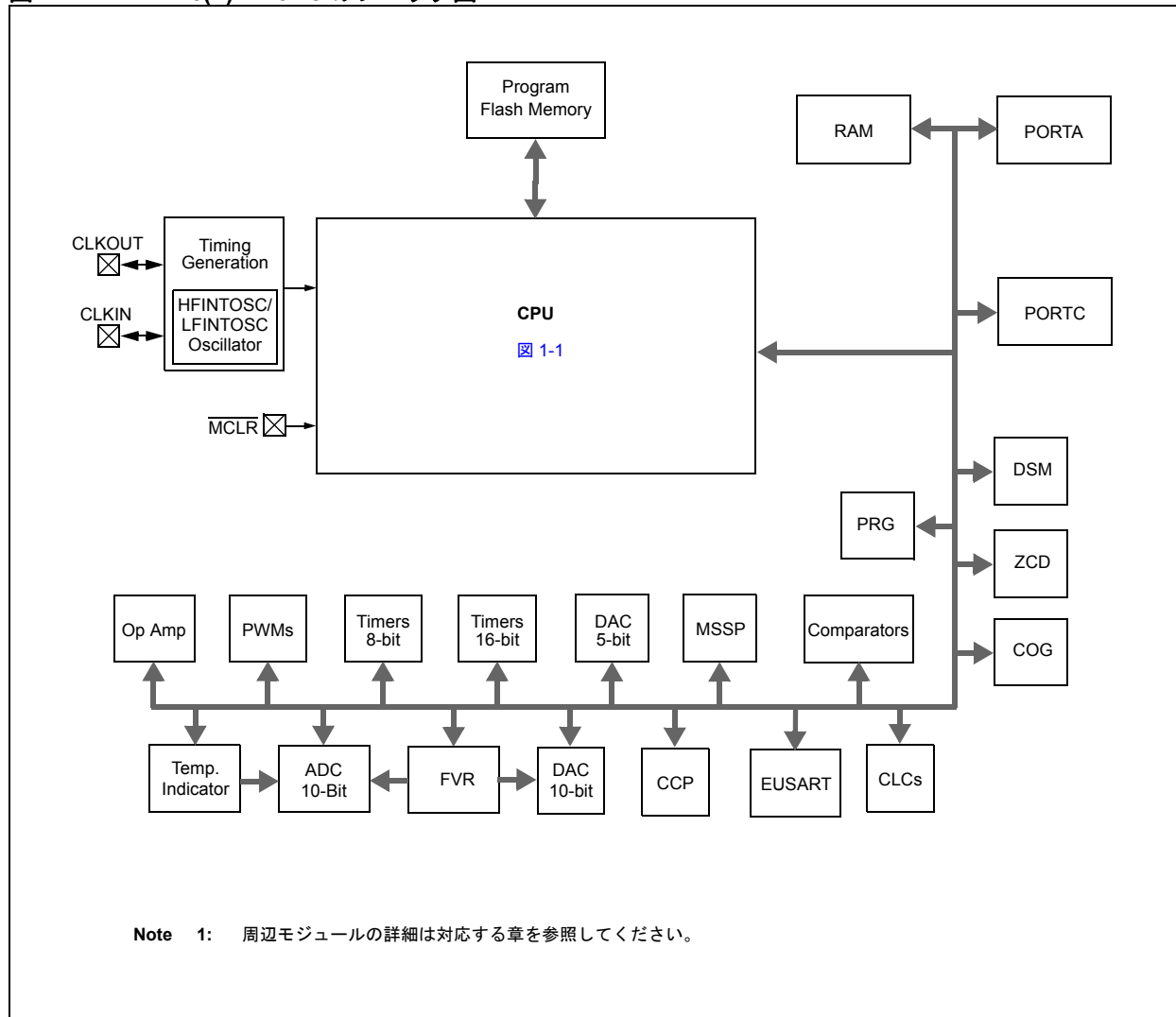
1.1.3.2 レガシー周辺モジュール

このような命名法に厳密には従わない周辺モジュールもあります。以前から使われており、今でもほとんど全てのデバイスに実装されている周辺モジュールです。このようなモジュール名は、レガシーコードとの互換性を維持するために必要でした。新規の命名法に従う周辺モジュールは、各周辺モジュール インスタンスに対する完全名接頭辞を示すテーブルをレジスタ部分に格納しています。上記の例外に含まれる周辺モジュールはこのテーブルを持っていません。そのような周辺モジュールの代表的なものは以下の通りです。

- EUSART
- MSSP

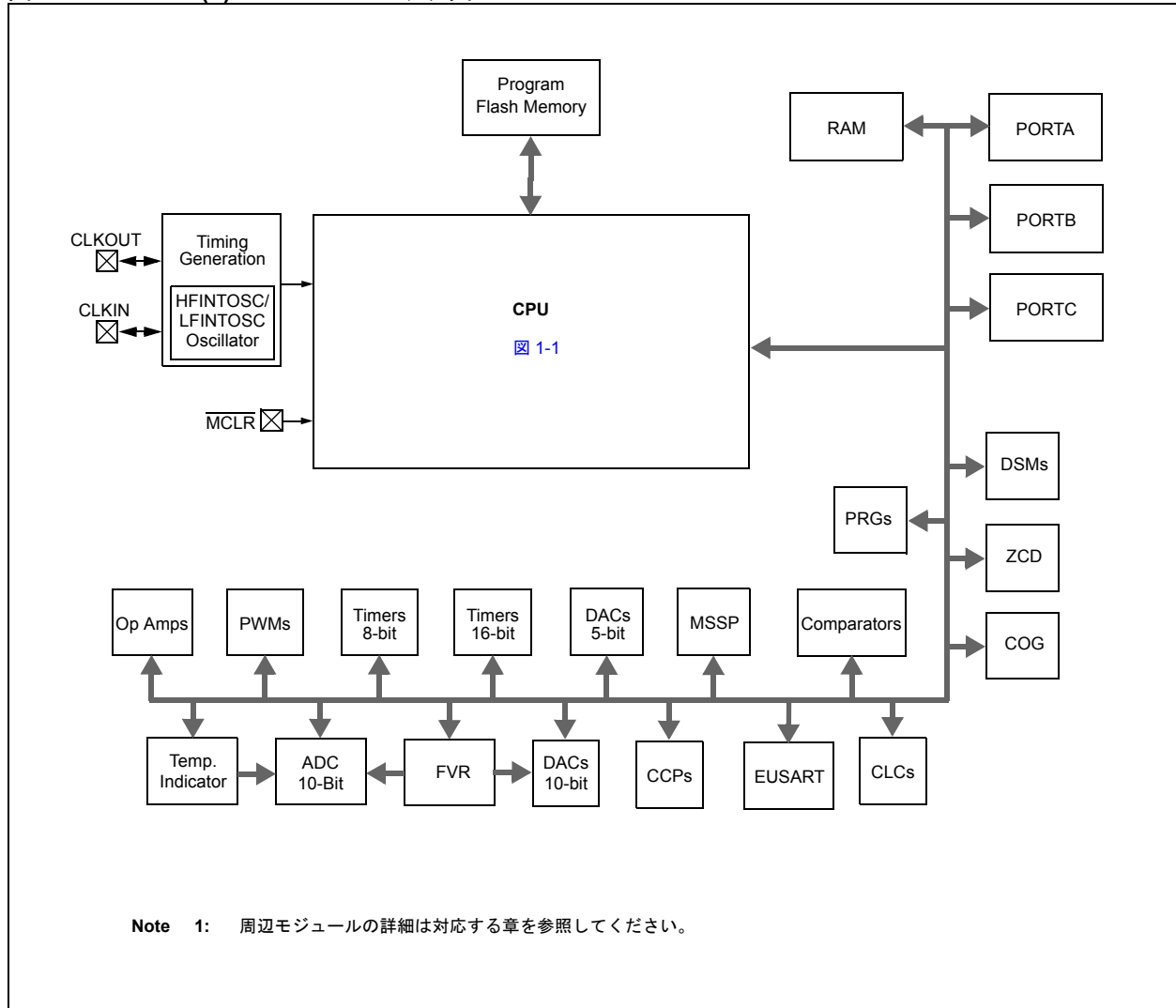
PIC16(L)F1764/5/8/9

図 1-1: PIC16(L)F1764/5 のブロック図



Note 1: 周辺モジュールの詳細は対応する章を参照してください。

図 1-2: PIC16(L)F1768/9 のブロック図



PIC16(L)F1764/5/8/9

表 1-2: PIC16(L)F1764/5 のピン割り当て

名前	機能	入力 タイプ	出力 タイプ	説明
RA0/AN0/C1IN0+/VREF-/ DAC1REF-/DAC3REF-/ DAC1OUT1/DAC3OUT1/ ICSPDAT	RA0	TTL/ST	CMOS	汎用 I/O
	AN0	AN	—	ADC チャンネル 0 入力
	C1IN0+	AN	—	コンパレータ C1 正入力
	VREF-	AN	—	ADC 負参照電圧
	DAC1REF-	AN	—	DAC1 負参照電圧
	DAC3REF-	AN	—	DAC3 負参照電圧
	DAC1OUT1	—	AN	DAC1 電圧出力
	DAC3OUT1	—	AN	DAC3 電圧出力
ICSPDAT	ST	CMOS	ICSP™ データ I/O	
RA1/AN1/C1IN0-/C2IN0-/VREF+/ DAC1REF+/DAC3REF+/ ICSPCLK	RA1	TTL/ST	CMOS	汎用 I/O
	AN1	AN	—	ADC チャンネル 1 入力
	C1IN0-	AN	—	コンパレータ C1 負入力
	C2IN0-	AN	—	コンパレータ C2 負入力
	VREF+	AN	—	ADC 正参照電圧
	DAC1REF+	AN	—	DAC1 正参照電圧
	DAC3REF+	AN	—	DAC3 正参照電圧
	ICSPCLK	ST	—	シリアル プログラミング クロック
RA2/AN2/ZCD/T0CKI/COG1IN/ INT	RA2	TTL/ST	CMOS	汎用 I/O
	AN2	AN	—	ADC チャンネル 2 入力
	ZCD	AN	—	ゼロクロス検出入力
	T0CKI	TTL/ST	—	Timer0 クロック入力
	COG1IN ⁽¹⁾	TTL/ST	—	相補出力ジェネレータ 1 入力
INT ⁽¹⁾	TTL/ST	—	割り込み入力	
RA3/T6CKI/MD1CH/MCLR/VPP	RA3	TTL/ST	CMOS	汎用 I/O
	T6CKI ⁽¹⁾	TTL/ST	—	Timer6 クロック入力
	MD1CH ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 1 搬送波 High 入力
	MCLR	ST	—	マスタクリア入力
	VPP	HV	—	プログラミング イネーブル
RA4/AN3/SOSCO/T1G/ MD1CL/OSC2/CLKOUT	RA4	TTL/ST	CMOS	汎用 I/O
	AN3	AN	—	ADC チャンネル 3 入力
	SOSCO	—	XTAL	セカンダリ オシレータ接続
	T1G ⁽¹⁾	TTL/ST	—	Timer1 ゲート入力
	MD1CL ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 1 搬送波 Low 入力
	OSC2	—	XTAL	水晶振動子 / セラミック振動子 (LP、XT、HS モード)
CLKOUT	—	CMOS	Fosc/4 出力	

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

表 1-2: PIC16(L)F1764/5 のピン割り当て (続き)

名前	機能	入力 タイプ	出力 タイプ	説明
RA5/T1CKI/T2CKI/CLCIN3/ MD1MOD/SOSCI/OSC1/CLKIN	RA5	TTL/ST	CMOS	汎用 I/O
	T1CKI ⁽¹⁾	TTL/ST	—	Timer1 クロック入力
	T2CKI ⁽¹⁾	TTL/ST	—	Timer2 クロック入力
	CLCIN3 ⁽¹⁾	TTL/ST	—	CLC 入力 3
	MD1MOD ⁽¹⁾	TTL/ST	—	データ信号モジュレータ変調入力
	SOSCI	—	XTAL	セカンダリ オシレータ接続
	OSC1	XTAL	—	水晶振動子 / セラミック振動子 (LP、XT、HS モード)
CLKIN	ST	—	外部クロック入力 (EC モード)	
RC0/AN4/OPA1IN+/C2IN0+/ T5CKI/SCL/SCK	RC0	TTL/ST	CMOS	汎用 I/O
	AN4	AN	—	ADC チャンネル 4 入力
	OPA1IN+	AN	—	オペアンプ 1 非反転入力
	C2IN0+	AN	—	コンパレータ 2 正入力
	T5CKI ⁽¹⁾	TTL/ST	—	Timer5 クロック入力
	SCL ^(1,3)	I ² C™	—	I ² C™ クロック出力
	SCK ⁽¹⁾	TTL/ST	—	SPI クロック入力
RC1/AN5/OPA1IN-/C1IN1-/ C2IN1-/T4CKI/CLCIN2/SDI/SDA	RC1	TTL/ST	CMOS	汎用 I/O
	AN5	AN	XTAL	ADC チャンネル 5 入力
	OPA1IN-	AN	—	オペアンプ 1 反転入力
	C1IN1-	AN	—	コンパレータ 1 負入力
	C2IN1-	AN	—	コンパレータ 2 負入力
	T4CKI ⁽¹⁾	TTL/ST	—	Timer4 クロック入力
	CLCIN2 ⁽¹⁾	TTL/ST	—	CLC 入力 2
	SDI ⁽¹⁾	TTL/ST	—	SPI データ入力
SDA ⁽¹⁾	I ² C™	—	I ² C™ データ出力	
RC2/AN6/OPA1OUT/C1IN2-/ C2IN2-/PRG1IN0	RC2	TTL/ST	CMOS	汎用 I/O
	AN6	AN	—	ADC チャンネル 6 入力
	OPA1OUT	—	AN	オペアンプ 1 出力
	C1IN2-	AN	—	コンパレータ 1 負入力
	C2IN2-	AN	—	コンパレータ 2 負入力
	PRG1IN0	AN	—	ランプ ジェネレータ 1 参照電圧入力
RC3/AN7/C1IN3-/C2IN3-/T5G/ CLCIN0/SS	RC3	TTL/ST	CMOS	汎用 I/O
	AN7	AN	—	ADC チャンネル 7 入力
	C1IN3-	AN	—	コンパレータ 1 負入力
	C2IN3-	AN	—	コンパレータ 2 負入力
	T5G ⁽¹⁾	TTL/ST	—	Timer5 ゲート入力
	CLCIN0 ⁽¹⁾	TTL/ST	—	CLC 入力 0
	SS ⁽¹⁾	TTL/ST	—	SPI スレーブ選択入力

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C™ レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

PIC16(L)F1764/5/8/9

表 1-2: PIC16(L)F1764/5 のピン割り当て (続き)

名前	機能	入力 タイプ	出力 タイプ	説明
RC4/T3G/PRG1R/CLCIN1/CK	RC4	TTL/ST	CMOS	汎用 I/O
	T3G ⁽¹⁾	TTL/ST	—	Timer3 ゲート入力
	PRG1R ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ set_rising 入力
	CLCIN1 ⁽¹⁾	TTL/ST	—	CLC 入力 1
	CK ⁽¹⁾	TTL/ST	—	EUSART クロック入力
RC5/T3CKI/PRG1F/CCP1/RX	RC5	TTL/ST	CMOS	汎用 I/O
	T3CKI ⁽¹⁾	TTL/ST	—	Timer3 クロック入力
	PRG1F ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ set_falling 入力
	CCP1 ⁽¹⁾	TTL/ST	—	CCP1 キャプチャ入力
	RX ^(1,3)	TTL/ST	—	EUSART 受信入力
VDD	VDD	電源	—	正電源
VSS	VSS	電源	—	参照グラウンド (GND)
OUT ⁽²⁾	C1OUT		CMOS	コンパレータ 1 出力
	C2OUT		CMOS	コンパレータ 2 出力
	CCP1		CMOS	コンペア /PWM1 出力
	MD1OUT		CMOS	データ信号モジュレータ 1 出力
	PWM3		CMOS	PWM3 出力
	PWM5		CMOS	PWM5 出力
	COG1A		CMOS	相補出力ジェネレータ出力 A
	COG1B		CMOS	相補出力ジェネレータ出力 B
	COG1C		CMOS	相補出力ジェネレータ出力 C
	COG1D		CMOS	相補出力ジェネレータ出力 D
	SDA ⁽³⁾		OD	I ² C™ データ出力
	SCK		CMOS	SPI クロック出力
	SCL ⁽³⁾		OD	I ² C™ クロック出力
	SDO		CMOS	SPI データ出力
	TX		CMOS	EUSART 非同期 TX データ出力
	CK		CMOS	EUSART 同期クロック出力
	DT ⁽³⁾		CMOS	EUSART 同期データ出力
	CLC1OUT		CMOS	構成可能なロジックセル 1 出力
	CLC2OUT		CMOS	構成可能なロジックセル 2 出力
	CLC3OUT		CMOS	構成可能なロジックセル 3 出力

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

表 1-3: PIC16(L)F1768/9 のピン割り当て

名前	機能	入力 タイプ	出力 タイプ	説明
RA0/AN0/C1IN0+/C3IN0+/VREF-/ DAC1REF-/DAC2REF-/ DAC3REF-/DAC4REF-/ DAC1OUT1/DAC2OUT1/ DAC3OUT1/DAC4OUT1/ ICSPDAT	RA0	TTL/ST	CMOS	汎用 I/O
	AN0	AN	—	ADC チャンネル 0 入力
	C1IN0+	AN	—	コンパレータ C1 正入力
	C3IN0+	AN	—	コンパレータ C3 正入力
	DAC1REF-	AN	—	DAC1 負参照電圧
	DAC2REF-	AN	—	DAC2 負参照電圧
	DAC3REF-	AN	—	DAC3 負参照電圧
	DAC4REF-	AN	—	DAC4 負参照電圧
	DAC1OUT1	—	AN	DAC1 電圧出力
	DAC2OUT1	—	AN	DAC2 電圧出力
	DAC3OUT1	—	AN	DAC3 電圧出力
	DAC4OUT1	—	AN	DAC4 電圧出力
	VREF-	AN	—	ADC 負参照電圧
	ICSPDAT	ST	CMOS	ICSP™ データ I/O
RA1/AN1/C1IN0-/C2IN0-/ C3IN0-/C4IN0-/VREF+/ DAC1REF+/DAC2REF+/ DAC3REF+/DAC4REF+/ ICSPCLK	RA1	TTL/ST	CMOS	汎用 I/O
	AN1	AN	—	ADC チャンネル 1 入力
	C1IN0-	AN	—	コンパレータ C1 負入力
	C2IN0-	AN	—	コンパレータ C2 負入力
	C3IN0-	AN	—	コンパレータ C3 負入力
	C4IN0-	AN	—	コンパレータ C4 負入力
	DAC1REF+	AN	—	DAC1 正参照電圧
	DAC2REF+	AN	—	DAC2 正参照電圧
	DAC3REF+	AN	—	DAC3 正参照電圧
	DAC4REF+	AN	—	DAC4 正参照電圧
	VREF+	AN	—	ADC 正参照電圧
	ICSPCLK	ST	—	シリアル プログラミング クロック
RA2/AN2/ZCD/T0CKI/COG1IN/ COG2IN/INT	RA2	TTL/ST	CMOS	汎用 I/O
	AN2	AN	—	ADC チャンネル 2 入力
	ZCD	AN	—	ゼロクロス検出入力
	T0CKI ⁽¹⁾	TTL/ST	—	Timer0 クロック入力
	COG1IN ⁽¹⁾	TTL/ST	—	相補出力ジェネレータ 1 入力
	COG2IN ⁽¹⁾	TTL/ST	—	相補出力ジェネレータ 2 入力
	INT ⁽¹⁾	TTL/ST	—	割り込み入力
RA3/T6CKI/MD1CH/MD2CH/ MCLR/VPP	RA3	TTL/ST	CMOS	汎用 I/O
	T6CKI ⁽¹⁾	TTL/ST	—	Timer6 クロック入力
	MD1CH ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 1 搬送波 High 入力
	MD2CH ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 2 搬送波 High 入力
	MCLR	ST	—	マスタクリア入力
VPP	HV	—	プログラミング イネーブル	

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

PIC16(L)F1764/5/8/9

表 1-3: PIC16(L)F1768/9 のピン割り当て (続き)

名前	機能	入力 タイプ	出力 タイプ	説明
RA4/AN3/SOSCO/T1G/ DSM1CL/DSM2CL/OSC2/ CLKOUT	RA4	TTL/ST	CMOS	汎用 I/O
	AN3	AN	—	ADC チャンネル 3 入力
	SOSCO	—	XTAL	セカンダリ オシレータ接続
	T1G ⁽¹⁾	TTL/ST	—	Timer1 ゲート入力
	DSM1CL ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 1 搬送波 Low 入力
	DSM2CL ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 2 搬送波 Low 入力
	OSC2	—	XTAL	水晶振動子 / セラミック振動子 (LP、XT、HS モード)
CLKOUT	—	CMOS	Fosc/4 出力	
RA5/T1CKI/T2CKI/CLCIN3/ DSM1MOD/DSM2MOD/ SOSCI/OSC1/CLKIN	RA5	TTL/ST	CMOS	汎用 I/O
	T1CKI ⁽¹⁾	TTL/ST	—	Timer1 クロック入力
	T2CKI ⁽¹⁾	TTL/ST	—	Timer2 クロック入力
	CLCIN3 ⁽¹⁾	TTL/ST	—	CLC 入力 3
	DSM1MOD ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 1 変調入力
	DSM2MOD ⁽¹⁾	TTL/ST	—	データ信号モジュレータ 2 変調入力
	SOSCI	XTAL	—	セカンダリ オシレータ接続
	OSC1	XTAL	—	水晶振動子 / セラミック振動子 (LP、XT、HS モード)
CLKIN	ST	—	外部クロック入力 (EC モード)	
RB4/AN10/OPA1IN0-/SDI/SDA	RB4	TTL/ST	CMOS	汎用 I/O
	AN10	AN	—	ADC チャンネル 10 入力
	OPA1IN0-	AN	—	オペアンプ 1 反転入力
	SDI ⁽¹⁾	TTL/ST	—	SPI データ入力
	SDA ^(1,3)	I ² C™	—	I ² C™ データ出力
RB5/AN11/OPA1IN0+/RX	RB5	TTL/ST	CMOS	汎用 I/O
	AN11	AN	—	ADC チャンネル 11 入力
	OPA1IN0+	AN	—	オペアンプ 1 非反転入力
	RX ^(1,3)	TTL/ST	—	EUSART 受信入力
RB6/C1IN1+/C3IN1+/SCK/SCL	RB6	TTL/ST	CMOS	汎用 I/O
	C1IN1+	AN	—	コンパレータ C1 正入力
	C3IN1+	AN	—	コンパレータ C3 正入力
	SCK ⁽¹⁾	TTL/ST	—	SPI クロック入力
	SCL ^(1,3)	I ² C™	—	I ² C™ クロック出力
RB7/C2IN1+/C4IN1+/CK	RB7	TTL/ST	CMOS	汎用 I/O
	C2IN1+	AN	—	コンパレータ C2 正入力
	C4IN1+	AN	—	コンパレータ C4 正入力
	CK ⁽¹⁾	TTL/ST	—	EUSART クロック入力
RC0/AN4/C2IN0+/C4IN0+/ T5CKI	RC0	TTL/ST	CMOS	汎用 I/O
	AN4	AN	—	ADC チャンネル 4 入力
	C2IN0+	AN	—	コンパレータ C2 正入力
	C4IN0+	AN	—	コンパレータ C4 正入力
	T5CKI ⁽¹⁾	TTL/ST	—	Timer5 クロック入力

凡例: AN = アナログ入力出力 CMOS = CMOS 互換入力出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

PIC16(L)F1764/5/8/9

表 1-3: PIC16(L)F1768/9 のピン割り当て (続き)

名前	機能	入力 タイプ	出力 タイプ	説明
RC1/AN5/C1IN1-/C2IN1-/ C3IN1-/C4IN1-/T4CKI/CLCIN2	RC1	TTL/ST	CMOS	汎用 I/O
	AN5	AN	XTAL	ADC チャンネル 5 入力
	C1IN1-	AN	—	コンパレータ 1 負入力
	C2IN1-	AN	—	コンパレータ 2 負入力
	C3IN1-	AN	—	コンパレータ 3 負入力
	C4IN1-	AN	—	コンパレータ 4 負入力
	T4CKI ⁽¹⁾	TTL/ST	—	Timer4 クロック入力
RC2/AN6/OPA1OUT/OPA2IN1-/ OPA2IN1+/C1IN2-/C2IN2-/ PRG1IN0/PRG2IN1	RC2	TTL/ST	CMOS	汎用 I/O
	AN6	AN	—	ADC チャンネル 6 入力
	OPA1OUT	—	AN	オペアンプ 1 出力
	OPA2IN1-	AN	—	オペアンプ 2 反転入力
	OPA2IN1+	AN	—	オペアンプ 2 非反転入力
	C1IN2-	AN	—	コンパレータ 1 負入力
	C2IN2-	AN	—	コンパレータ 2 負入力
	PRG1IN0	AN	—	ランプ ジェネレータ 1 参照電圧入力
	PRG2IN1	AN	—	ランプ ジェネレータ 2 参照電圧入力
RC3/AN7/OPA2OUT/OPA1IN1-/ OPA1IN1+/C1IN3-/C2IN3-/ C3IN3-/C4IN3-/PRG1IN1/ PRG2IN0/T5G/CCP2/CLCIN0	RC3	TTL/ST	CMOS	汎用 I/O
	AN7	AN	—	ADC チャンネル 7 入力
	OPA2OUT	—	AN	オペアンプ 2 出力
	OPA1IN1-	AN	—	オペアンプ 1 反転入力
	OPA1IN1+	AN	—	オペアンプ 1 非反転入力
	C1IN3-	AN	—	コンパレータ 1 負入力
	C2IN3-	AN	—	コンパレータ 2 負入力
	C3IN3-	AN	—	コンパレータ 3 負入力
	C4IN3-	AN	—	コンパレータ 4 負入力
	PRG1IN1	AN	—	ランプ ジェネレータ 1 参照電圧入力
	PRG2IN0	AN	—	ランプ ジェネレータ 2 参照電圧入力
	T5G ⁽¹⁾	TTL/ST	—	Timer5 ゲート入力
	CCP2 ⁽¹⁾	TTL/ST	—	CCP2 キャプチャ入力
	CLCIN0 ⁽¹⁾	TTL/ST	—	CLC 入力 0
RC4/T3G/PRG1R/PRG2R/ CLCIN1	RC4	TTL/ST	CMOS	汎用 I/O
	T3G ⁽¹⁾	TTL/ST	—	Timer3 ゲート入力
	PRG1R ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ 1 set_rising 入力
	PRG2R ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ 2 set_rising 入力
	CLCIN1 ⁽¹⁾	TTL/ST	—	CLC 入力 1
RC5/T3CKI/PRG1F/PRG2F/ CCP1	RC5	TTL/ST	CMOS	汎用 I/O
	T3CKI ⁽¹⁾	TTL/ST	—	Timer3 クロック入力
	PRG1F ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ 1 set_falling 入力
	PRG2F ⁽¹⁾	TTL/ST	—	ランプ ジェネレータ 2 set_falling 入力
	CCP1 ⁽¹⁾	TTL/ST	—	CCP1 キャプチャ入力

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

PIC16(L)F1764/5/8/9

表 1-3: PIC16(L)F1768/9 のピン割り当て (続き)

名前	機能	入力 タイプ	出力 タイプ	説明
RC6/AN8/OPA2IN0-SS	RC6	TTL/ST	CMOS	汎用 I/O
	AN8	AN	—	ADC チャンネル 8 入力
	OPA2IN0-	AN	—	オペアンプ 2 反転入力
	SS ⁽¹⁾	TTL/ST	—	SPI スレーブ選択入力
RC7/AN9/OPA2IN0+	RC7	TTL/ST	CMOS	汎用 I/O
	AN9	AN	—	ADC チャンネル 9 入力
	OPA2IN0+	AN	—	オペアンプ 2 非反転入力
VDD	VDD	電源	—	正電源
VSS	VSS	電源	—	参照グラウンド (GND)
OUT ⁽²⁾	C1OUT		CMOS	コンパレータ 1 出力
	C2OUT		CMOS	コンパレータ 2 出力
	C3OUT		CMOS	コンパレータ 3 出力
	C4OUT		CMOS	コンパレータ 4 出力
	CCP1		CMOS	コンペア /PWM1 出力
	CCP2		CMOS	コンペア /PWM2 出力
	MD1OUT		CMOS	データ信号モジュレータ 1 出力
	MD2OUT		CMOS	データ信号モジュレータ 2 出力
	PWM3		CMOS	PWM3 出力
	PWM4		CMOS	PWM4 出力
	PWM5		CMOS	PWM5 出力
	PWM6		CMOS	PWM6 出力
	COG1A		CMOS	相補出力ジェネレータ 1 出力 A
	COG1B		CMOS	相補出力ジェネレータ 1 出力 B
	COG1C		CMOS	相補出力ジェネレータ 1 出力 C
	COG1D		CMOS	相補出力ジェネレータ 1 出力 D
	COG2A		CMOS	相補出力ジェネレータ 2 出力 A
	COG2B		CMOS	相補出力ジェネレータ 2 出力 B
	COG2C		CMOS	相補出力ジェネレータ 2 出力 C
	COG2D		CMOS	相補出力ジェネレータ 2 出力 D
	SDA ⁽³⁾		OD	I ² C™ データ出力
	SCK		CMOS	SPI クロック出力
	SCL ⁽³⁾		OD	I ² C™ クロック出力
	SDO		CMOS	SPI データ出力
	TX		CMOS	EUSART 非同期 TX データ出力
	CK		CMOS	EUSART 同期クロック出力
	DT ⁽³⁾		CMOS	EUSART 同期データ出力
	CLC1OUT		CMOS	構成可能なロジックセル 1 出力
	CLC2OUT		CMOS	構成可能なロジックセル 2 出力
	CLC3OUT		CMOS	構成可能なロジックセル 3 出力

凡例: AN = アナログ入出力 CMOS = CMOS 互換入出力 OD = オープンドレイン
TTL = TTL 互換入力 ST = CMOS レベルのシュミットトリガ入力 I²C™ = I²C レベルのシュミットトリガ入力
HV = 高電圧 (High Voltage) XTAL = 水晶振動子レベル

- Note** 1: 周辺モジュールの既定値入力です。PPS 入力選択レジスタを使うと、周辺モジュールの入力として別のピンを選択できます。
- 2: 全てのピンのデジタル出力の既定値は、PORT ラッチデータです。PPS 出力選択レジスタを使うと、周辺モジュールのデジタル出力として別の出力を選択できます。
- 3: これらの周辺機能は双方向です。出力ピンの選択は、入力ピンの選択と同じである必要があります。

1.2 周辺モジュール接続マトリクス

多くの周辺モジュールが備える入力選択マルチプレクサを使うと、別の周辺モジュールの出力を入力として選択でき、信号経路全体をデバイス内に収める事ができます。周辺モジュールの出力は PPS 機能を使ってピンに出す事もできます。表 1-4 に、周辺モジュール間で可能な信号接続を示します。特定の接続のマルチプレクサ選択コードを知るには、対応する周辺モジュールのセクションを参照してください。

PIC16(L)F1764/5/8/9

表 1-4: 周辺モジュール接続マトリクス

周辺モジュールの出力	周辺モジュールの入力																									
	ADC トリガ	COG クロック	COG 立ち上がり / 立ち下がり	COG シャットダウン	10 ビット DAC	5 ビット DAC	PRG アナログ入力	PRG 立ち上がり / 立ち下がり	コンパレータ (+)	コンパレータ (-)	CLC	DSM CH	DSM CL	DSM Mod	オペアンプ (+)	オペアンプ (-)	オペアンプオーバーライド	10 ビット PWM	16 ビット PWM	CCP キャプチャ	CCP クロック	Timer2/4/6 クロック	Timer2/4/6 リセット	Timer1/3/5 ゲート	Timer0 クロック	
FVR					•	•	•		•	•					•	•										
ZCD											•						•						•			
PRG									•						•	•										
10 ビット DAC									•						•	•										
5 ビット DAC									•						•	•										
CCP	•		•					•			•	•	•	•			•									
コンパレータ (同期)	•							•			•						•			•				•		•
コンパレータ (非同期)			•	•										•												
CLC	•		•	•							•	•	•	•			•			•			•	•		
DSM																										
COG																	•									
EUSART TX/CK											•			•												
EUSART DT											•			•												
MSSP SCK/SCL											•			•												
MSSP SDO/SDA											•			•												
オペアンプ									•																	
10 ビット PWM	•		•					•			•	•	•	•			•							•		
16 ビット PWM	•		•					•			•	•	•	•			•							•		
Timer0 オーバーフロー	•										•														•	
Timer2 = T2PR				•							•								•			•		•		
Timer4 = T4PR				•							•								•			•		•		
Timer6 = T6PR				•							•								•			•		•		
Timer2 ポストスケール	•			•							•								•			•		•		
Timer4 ポストスケール	•			•							•								•			•		•		
Timer6 ポストスケール	•			•							•								•			•		•		
Timer1 オーバーフロー	•										•								•			•		•		
Timer3 オーバーフロー	•										•								•			•		•		
Timer5 オーバーフロー	•										•								•			•		•		
SOSC																			•				•			
Fosc/4		•																						•		
Fosc		•									•	•	•						•					•		
HFINTOSC		•									•	•	•						•					•		
LFINTOSC											•								•					•		
MFINTOSC											•								•					•		
IOCIF											•									•	•					
PPS 入力ピン			•	•				•			•	•	•						•	•	•	•	•	•	•	•

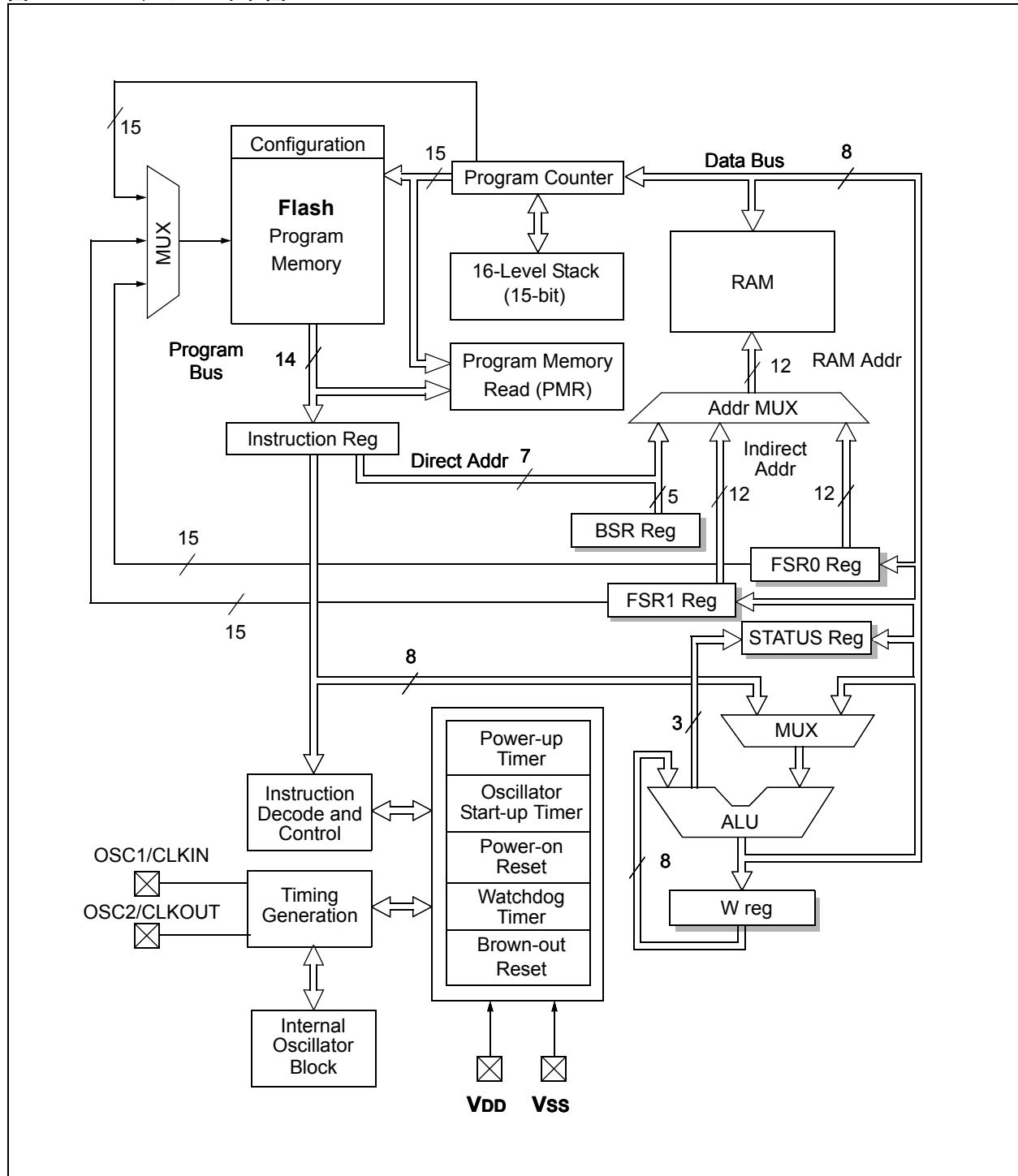
2.0 エンハスト ミッドレンジ CPU

本デバイスファミリはエンハスト ミッドレンジ 8 ビット CPU コアを採用しています。CPU は 49 種類の命令を実行できます。割り込みは自動コンテキスト保存機能を備えています。ハードウェアスタックの深さは 16 段であり、オーバーフローとアンダーフローによるリセット機能を備えています。アドレス指定

モードは、直接、間接、相対をサポートします。2 つの FSR(ファイルセレクト レジスタ) を使ってプログラムメモリとデータメモリの読み出しが可能です。

- 割り込み発生時の自動コンテキスト保存
- オーバーフロー/アンダーフロー リセット機能付き 16 段スタック
- FSR(ファイルセレクト レジスタ)
- 命令セット

図 2-1: コアのブロック図



PIC16(L)F1764/5/8/9

2.1 割り込み発生時の自動コンテキスト保存

割り込み発生時に一部のレジスタ内容を自動的にシャドウレジスタに保存し、通常動作に戻る際に復元できます。これにより、スタック空間とユーザコードを節約できます。詳細は[セクション 7.5「コンテキスト自動保存機能」](#)を参照してください。

2.2 オーバーフロー/アンダーフローリセット機能付き 16 段スタック

本デバイスにはハードウェア スタックメモリがあります(幅 15 ビット x 深さ 16 ワード)。スタック オーバーフローやアンダーフローが生じると PCON レジスタで対応するビット STKOVF または STKUNF がセットされ、ソフトウェア リセットが有効の場合はソフトウェア リセットが発生します。詳細は[セクション 3.6「スタック」](#)を参照してください。

2.3 FSR(ファイルセレクト レジスタ)

16 ビットの FSR が 2 つあります。これらの FSR は全てのファイルレジスタとプログラムメモリにアクセスでき、全てのメモリに対して同じデータポインタを使う事ができます。FSR を使ってプログラムメモリへアクセスする場合、INDF レジスタへアクセスしてデータフェッチする命令では追加で 1 命令サイクルが必要です。これにより、汎用メモリへのリニアなアドレス指定が可能で、80 バイトを超える連続したデータにもアクセスできます。FSR をサポートする新しい命令もあります。詳細は[セクション 3.7「間接アドレス指定」](#)を参照してください。

2.4 命令セット

エンハンスド ミッドレンジ CPU には CPU 機能をサポートする 49 個の命令があります。詳細は[セクション 35.0「命令セットのまとめ」](#)を参照してください。

3.0 メモリの構成

これらのデバイスは以下のタイプのメモリを搭載しています。

- プログラムメモリ
 - コンフィグレーションワード
 - デバイス ID
 - ユーザ ID
 - フラッシュ プログラムメモリ
- データメモリ
 - コアレジスタ
 - 特殊機能レジスタ
 - 汎用 RAM
 - 共通 RAM

Note 1: PMCON レジスタを使ったフラッシュメモリへのアクセス方法は[セクション 10.0「フラッシュ プログラムメモリ制御」](#)で説明します。

プログラムメモリとデータメモリのアクセスと制御に関する機能として、以下のものがあります。

- PCL と PCLATH
- スタック
- 間接アドレス指定

表 3-1: デバイスサイズとアドレス

デバイス	プログラムメモリ空間 (ワード数)	プログラムメモリの 最終アドレス	高書き込み耐性フラッシュメモリの アドレス範囲 ⁽¹⁾
PIC16(L)F1764	4,096	0FFFh	0F80h ~ 0FFFh
PIC16(L)F1765	8,192	1FFFh	1F80h ~ 1FFFh
PIC16(L)F1768	4,096	0FFFh	0F80h ~ 0FFFh
PIC16(L)F1769	8,192	1FFFh	1F80h ~ 1FFFh

Note 1: 高書き込み耐性フラッシュは、範囲内の各アドレス下位バイトのみです。

3.1 プログラムメモリの構成

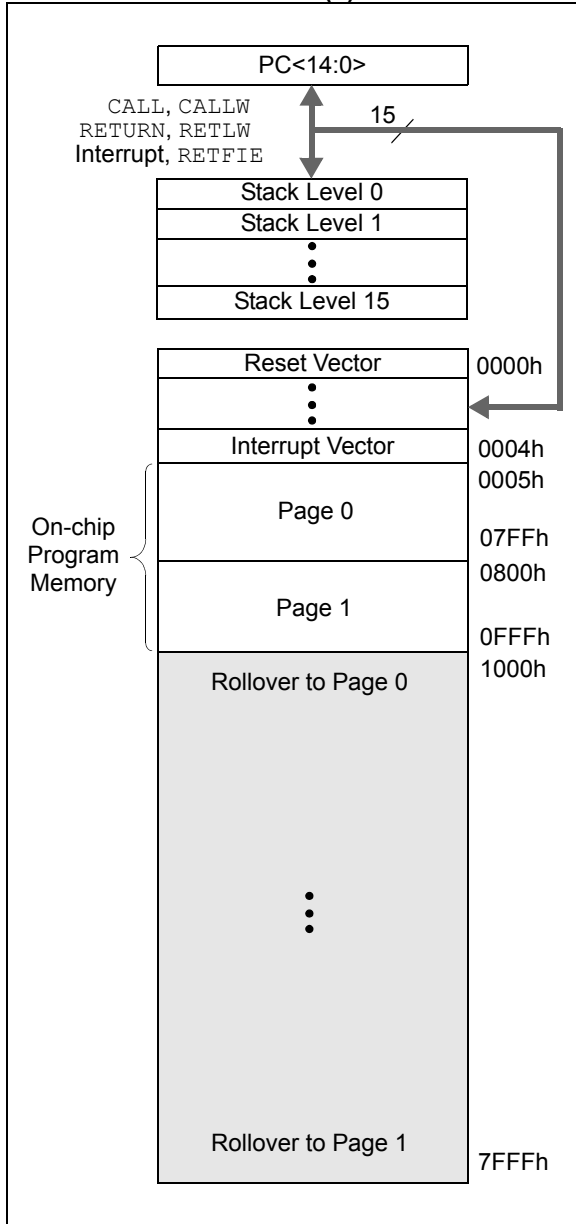
エンハンスド ミッドレンジコアには 15 ビットのプログラムカウンタがあり、32K x 14 のプログラムメモリ空間におけるアドレス指定が可能です。[表 3-1](#) に、PIC16(L)F1764/5/8/9ファミリに実装されているメモリサイズを示します。この境界を越える位置にアクセスすると、実装されたメモリ空間内でラップアラウンド (折り返し) が発生します。リセットベクタは 0000h、割り込みベクタは 0004h です ([図 3-1](#) 参照)。

3.2 高書き込み耐性フラッシュ

本デバイスはデータ EEPROM の代わりに 128 バイト分の高書き込み耐性プログラムフラッシュメモリ (PFM) を備えます。この領域は最終製品の寿命期間を通じて頻繁に更新される事が予測される不揮発性データストレージとして最適です。PFM へのデータの書き込みの詳細は[セクション 10.2「フラッシュ プログラムメモリの概要」](#)を参照してください。PFM に保存したバイトデータを FSR レジスタを使って読み出す方法の詳細は[セクション 3.2.1.2「FSR を使った間接読み出し」](#)を参照してください。

PIC16(L)F1764/5/8/9

図 3-1: のプログラムメモリ マップとメモリ スタック PIC16(L)F1764/5/8/9



3.2.1 プログラムメモリをデータとして読み出す

プログラムメモリの定数にアクセスする方法は 2 つあります。1 つ目は、RETLW 命令テーブルを使う方法です。2 つ目は、FSR を介してプログラムメモリ内を指定する方法です。

3.2.1.1 RETLW 命令

RETLW 命令を使って定数テーブルへアクセスできます。例 3-1 に、推奨するテーブル作成方法を示します。

例 3-1: RETLW 命令

```
constants
    BRW                ;Add Index in W to
                       ;program counter to
                       ;select data

    RETLW DATA0       ;Index0 data
    RETLW DATA1       ;Index1 data
    RETLW DATA2
    RETLW DATA3

my_function
    ;... LOTS OF CODE...
    MOVLW DATA_INDEX
    call constants
    ;... THE CONSTANT IS IN W
```

BRW 命令を使うと、このようなテーブルを非常に簡単に実装できます。旧世代のマイクロコントローラとのコード移植性を確保する必要がある場合、BRW 命令を使えないため、従来のテーブル読み出し方法を使う必要があります。

3.2.1.2 FSR を使った間接読み出し

FSRxH レジスタの bit 7 をセットし、一致する INDFx レジスタを読み出す事で、プログラムメモリへデータとしてアクセスできます。MOVLW 命令は、アドレス指定したワードの下位 8 ビットを W レジスタへ格納します。INDF レジスタを介してプログラムメモリへ書き込む事はできません。FSR を使ってプログラムメモリへアクセスする場合、追加で 1 命令サイクルが必要です。例 3-2 に FSR を介したプログラムメモリへのアクセスを示します。

ラベルがプログラムメモリ内の位置を指し示している場合、High ディレクティブにより bit<7> がセットされます。

例 3-2: FSR を介したプログラムメモリへのアクセス

```
constants
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants
    MOVWF FSR1H
    MOVIW 0「」FSR1」
;THE PROGRAM MEMORY IS IN W
```

3.3 データメモリの構成

データメモリは、32 個のメモリバンクに分割されます (1 バンクは 128 バイト)。各バンクは以下の内容で構成されます (図 3-2 参照)。

- 12 個のコアレジスタ
- 20 個の特殊機能レジスタ (SFR)
- 最大 80 バイトの汎用 RAM (GPR)
- 16 バイトの共通 RAM

バンクセレクトレジスタ (BSR) にバンク番号を書き込む事で、アクティブバンクを選択します。未実装のメモリは「0」として読み出されます。全てのデータメモリへのアクセスは、直接的 (ファイルレジスタを使う命令を使用) または間接的 (2 つのファイルセレクトレジスタ (FSR) を使用) に実行できます。詳細は [セクション 3.7 「間接アドレス指定」](#) を参照してください。

データメモリは 12 ビットのアドレスを使います。アドレスの上位 5 ビットはバンクアドレスを定義し、下位 7 ビットはそのバンク内のレジスタ /RAM を選択します。

3.3.1 コアレジスタ

コアレジスタには、基本動作に直接影響を与えるレジスタが含まれます。コアレジスタは各データメモリバンクの最初の 12 アドレスを占有しています (アドレス x00h/x08h ~ x0Bh/x8Bh)。表 3-2 に、それらのレジスタを示します。詳細は表 3-15 を参照してください。

表 3-2: コアレジスタ

アドレス	BANKx
x00h または x80h	INDF0
x01h または x81h	INDF1
x02h または x82h	PCL
x03h または x83h	STATUS
x04h または x84h	FSR0L
x05h または x85h	FSR0H
x06h または x86h	FSR1L
x07h または x87h	FSR1H
x08h または x88h	BSR
x09h または x89h	WREG
x0Ah または x8Ah	PCLATH
x0Bh または x8Bh	INTCON

3.3.1.1 STATUS レジスタ

STATUS レジスタ (レジスタ 3-1 参照) の内容は、以下の通りです。

- ALU の演算状態
- リセット状態

STATUS レジスタは、他の全てのレジスタと同様に任意の命令の格納先とする事ができます。STATUS レジスタが Z、DC、C のいずれかのビットに影響を及ぼす命令の格納先である場合、これら 3 つのビットには書き込みできません。これらのビットはデバイスのロジックに従ってセットまたはクリアされます。また、TO ビットと PD ビットには書き込みできません。従って、STATUS レジスタを格納先とする命令を実行した場合、意図した結果とならない場合があります。

例えば、CLRF STATUS は上位 3 ビットをクリアし、Z ビットをセットします。これにより、STATUS レジスタは「000u u1uu」(u= 不変) のままです。

従って、STATUS レジスタを変更する際は BCF、BSF、SWAPF、MOVWF 命令等、ステータスビットに影響を与えない命令を推奨します。ステータスビットに影響を与えないその他の命令については、[セクション 35.0 「命令セットのまとめ」](#) を参照してください。

Note: 減算では、C ビットが Borrow、DC ビットが Digit Borrow アウトビットとして動作します。

PIC16(L)F1764/5/8/9

3.4 レジスタ定義 : STATUS

レジスタ 3-1: STATUS: STATUS レジスタ

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	TO	PD	Z	DC ⁽¹⁾	C ⁽¹⁾
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7-5 **未実装:** 「0」として読み出し

bit 4 **TO:** タイムアウト ビット

- 1 = 電源投入後、CLRWDT 命令後、SLEEP 命令後のいずれか
- 0 = WDT タイムアウトが発生した

bit 3 **PD:** パワーダウン ビット

- 1 = 電源投入後、または CLRWDT 命令後のいずれか
- 0 = SLEEP 命令を実行した

bit 2 **Z:** ゼロビット

- 1 = 算術演算または論理演算の結果がゼロである
- 0 = 算術演算または論理演算の結果はゼロでない

bit 1 **DC:** Digit Carry/Digit Borrow ビット (ADDWF、ADDLW、SUBLW、SUBWF 命令)⁽¹⁾

- 1 = 結果の最下位から 4 ビット目でキャリーアウトが発生した
- 0 = 結果の最下位から 4 ビット目でキャリーアウトが発生していない

bit 0 **C:** Carry/Borrow ビット⁽¹⁾ (ADDWF、ADDLW、SUBLW、SUBWF 命令)⁽¹⁾

- 1 = 演算結果の最上位ビットからキャリーアウトが発生した
- 0 = 演算結果の最上位ビットからキャリーアウトは発生していない

Note 1: Borrow の場合、極性は逆です。減算は、2 番目のオペランドの 2 の補数を加算する事で実行します。

3.4.1 特殊機能レジスタ

特殊機能レジスタ (SFR) は、デバイス内の周辺モジュールを動作させるためにアプリケーションが使うレジスタです。SFR は各データメモリバンクのコアレジスタに続く 20 バイトを占有しています (アドレス x0Ch/x8Ch ~ x1Fh/x9Fh)。各周辺モジュールに関連するレジスタについては、対応する周辺モジュールの章で説明します。

3.4.2 汎用 RAM

各データメモリバンクには、最大 80 バイトの汎用レジスタ (GPR) があります。GPR は選択されたデータメモリバンクの SFR の直後の空間を占めます。選択されるバンク数はデバイス内で利用可能な GPR 空間の合計によって異なります。

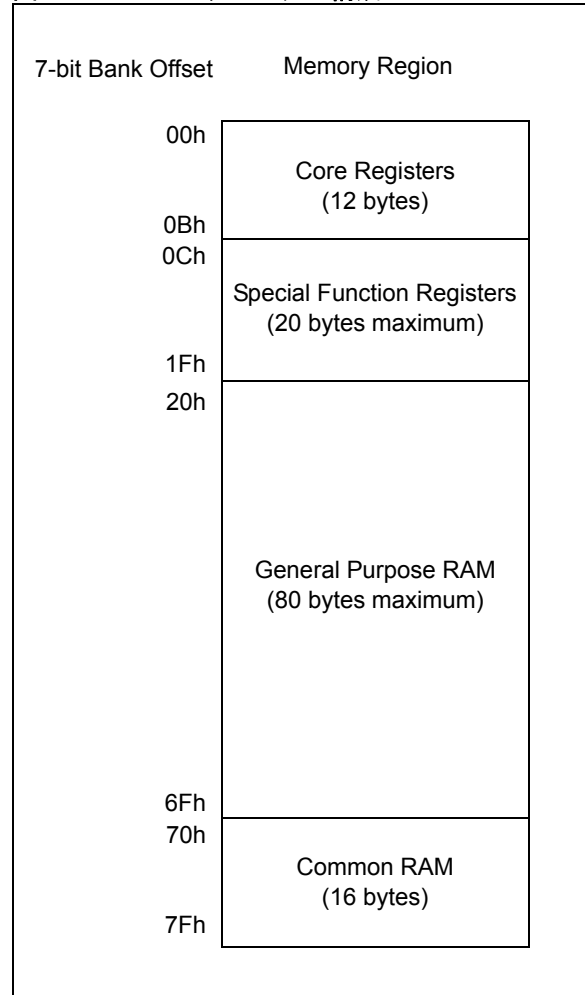
3.4.2.1 GPR へのリニアアクセス

FSR を使うと、バンク切り換えなしに汎用 RAM にアクセスできます。これにより、大規模なメモリ構造に容易にアクセスできます。詳細は[セクション 3.7.2「リニアデータメモリ」](#)を参照してください。

3.4.3 共通 RAM

全てのバンクからアクセス可能な 16 バイトの共通 RAM があります。

図 3-2: メモリバンクの構成



3.4.4 デバイスメモリマップ

表 3-3 から 3-14 に、本デバイスファミリのメモリマップを示します。

表 3-3: PIC16(L)F1764 のメモリマップ (BANK 0 ~ 7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7						
000h	コアレジスタ (表 3-2)	080h	コアレジスタ (表 3-2)	100h	コアレジスタ (表 3-2)	180h	コアレジスタ (表 3-2)	200h	コアレジスタ (表 3-2)	280h	コアレジスタ (表 3-2)	300h	コアレジスタ (表 3-2)	380h	コアレジスタ (表 3-2)					
00Bh	—	08Bh	—	10Bh	—	18Bh	—	20Bh	—	28Bh	—	30Bh	—	38Bh	—					
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA					
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—					
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC					
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—					
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	390h	—					
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP					
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN					
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF					
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	—					
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	—					
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	—					
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP					
018h	T1CON	098h	OSCTUNE	118h	—	198h	—	218h	—	298h	—	318h	—	398h	IOCCN					
019h	T1GCON	099h	OSCCON	119h	—	199h	RC1REG	219h	—	299h	—	319h	—	399h	IOCCF					
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	—	31Ah	—	39Ah	—					
01Bh	T2PR	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	MD1CON0					
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	MD1CON1					
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	—	39Dh	MD1SRC					
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	—	39Eh	MD1CARL					
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	—	39Fh	MD1CARH					
020h	汎用レジスタ 80 バイト	0A0h	汎用レジスタ 80 バイト	120h	汎用レジスタ 80 バイト	1A0h	汎用レジスタ 80 バイト	220h	汎用レジスタ 80 バイト	2A0h	汎用レジスタ 80 バイト	320h	汎用レジスタ 16 バイト	3A0h	未実装 「0」として 読み出し					
06Fh		共通 RAM 70h ~ 7Fh		0EFh		アクセス 70h ~ 7Fh		16Fh		アクセス 70h ~ 7Fh		1EFh		アクセス 70h ~ 7Fh		26Fh	アクセス 70h ~ 7Fh	2EFh	アクセス 70h ~ 7Fh	36Fh
070h				—				0F0h				—	1F0h			—		2F0h		—
07Fh		—				0FFh		—		1FFh			—	2FFh			—	3FFh	—	

凡例: = 未実装のデータメモリ領域、「0」として読み出し

Note 1: PIC16LF1764 では未実装です。

表 3-4: PIC16LF1765 のメモリマップ (BANK 0 ~ 7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	コアレジスタ (表 3-2)	080h	コアレジスタ (表 3-2)	100h	コアレジスタ (表 3-2)	180h	コアレジスタ (表 3-2)	200h	コアレジスタ (表 3-2)	280h	コアレジスタ (表 3-2)	300h	コアレジスタ (表 3-2)	380h	コアレジスタ (表 3-2)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	—
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	—	198h	—	218h	—	298h	—	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	—	199h	RC1REG	219h	—	299h	—	319h	—	399h	IOCCF
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	T2PR	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	MD1CON0
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	MD1CON1
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	—	39Dh	MD1SRC
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	—	39Eh	MD1CARL
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	—	39Fh	MD1CARH
020h		0A0h		120h		1A0h		220h		2A0h		320h		3A0h	
	汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h		170h		1F0h		270h		2F0h		370h		3F0h	
	共通 RAM 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

凡例: = 未実装のデータメモリ領域、「0」として読み出し

Note 1: PIC16LF1765 では未実装です。

表 3-5: PIC16(L)F1768 のメモリマップ (BANK 0 ~ 7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7								
000h	コアレジスタ (表 3-2)	080h	コアレジスタ (表 3-2)	100h	コアレジスタ (表 3-2)	180h	コアレジスタ (表 3-2)	200h	コアレジスタ (表 3-2)	280h	コアレジスタ (表 3-2)	300h	コアレジスタ (表 3-2)	380h	コアレジスタ (表 3-2)							
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh								
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA							
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB							
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC							
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—							
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	390h	—							
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP							
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN							
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF							
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	IOCBP							
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN							
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF							
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP							
018h	T1CON	098h	OSCTUNE	118h	CM3CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN							
019h	T1GCON	099h	OSCCON	119h	CM3CON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	—	399h	IOCCF							
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	CM3NSEL	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—							
01Bh	T2PR	09Bh	ADRESL	11Bh	CM3PSEL	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	MD2CON0	39Bh	MD1CON0							
01Ch	T2CON	09Ch	ADRESH	11Ch	CM4CON0	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	MD2CON1	39Ch	MD1CON1							
01Dh	T2HLT	09Dh	ADCON0	11Dh	CM4CON1	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	MD2SRC	39Dh	MD1SRC							
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	CM4NSEL	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	MD2CARL	39Eh	MD1CARL							
01Fh	T2RST	09Fh	ADCON2	11Fh	CM4PSEL	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	MD2CARH	39Fh	MD1CARH							
020h	汎用レジスタ 80 バイト	0A0h	汎用レジスタ 80 バイト	120h	汎用レジスタ 80 バイト	1A0h	汎用レジスタ 80 バイト	220h	汎用レジスタ 80 バイト	2A0h	汎用レジスタ 80 バイト	320h	汎用レジスタ 16 バイト	3A0h	未実装 「0」として 読み出し							
06Fh				0EFh				16Fh				1EFh		26Fh			2EFh		36Fh			
070h		共通 RAM 70h ~ 7Fh		0F0h		アクセス 70h ~ 7Fh		170h		アクセス 70h ~ 7Fh		1F0h	アクセス 70h ~ 7Fh	270h		アクセス 70h ~ 7Fh	2F0h	アクセス 70h ~ 7Fh	370h	アクセス 70h ~ 7Fh	3F0h	アクセス 70h ~ 7Fh
07Fh								0FFh						17Fh					1FFh		27Fh	

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

Note 1: PIC16LF1768 では未実装です。

表 3-6: PIC16(L)F1769 のメモリマップ (BANK 0 ~ 7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	コアレジスタ (表 3-2)	080h	コアレジスタ (表 3-2)	100h	コアレジスタ (表 3-2)	180h	コアレジスタ (表 3-2)	200h	コアレジスタ (表 3-2)	280h	コアレジスタ (表 3-2)	300h	コアレジスタ (表 3-2)	380h	コアレジスタ (表 3-2)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	CM3CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	CM3CON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	—	399h	IOCCF
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	CM3NSEL	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	T2PR	09Bh	ADRESL	11Bh	CM3PSEL	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	MD2CON0	39Bh	MD1CON0
01Ch	T2CON	09Ch	ADRESH	11Ch	CM4CON0	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	MD2CON1	39Ch	MD1CON1
01Dh	T2HLT	09Dh	ADCON0	11Dh	CM4CON1	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	MD2SRC	39Dh	MD1SRC
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	CM4NSEL	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	MD2CARL	39Eh	MD1CARL
01Fh	T2RST	09Fh	ADCON2	11Fh	CM4PSEL	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	MD2CARH	39Fh	MD1CARH
020h		0A0h		120h		1A0h		220h		2A0h		320h		3A0h	
	汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト		汎用レジスタ 80 バイト
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h		170h		1F0h		270h		2F0h		370h		3F0h	
	共通 RAM 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh		アクセス 70h ~ 7Fh
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

Note 1: PIC16LF1769 では未実装です。

表 3-7: PIC16(L)F1764 のメモリマップ (BANK 8 ~ 23)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	コアレジスタ (表 3-2)	480h	コアレジスタ (表 3-2)	500h	コアレジスタ (表 3-2)	580h	コアレジスタ (表 3-2)	600h	コアレジスタ (表 3-2)	680h	コアレジスタ (表 3-2)	700h	コアレジスタ (表 3-2)	780h	コアレジスタ (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACL	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	—	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	—	594h	—	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	—	595h	—	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	—	596h	—	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DGH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	—	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Ah	—	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	—
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	—	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	—	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	—
420h	未実装 「0」として 読み出し	4A0h	未実装 「0」として 読み出し	520h	未実装 「0」として 読み出し	5A0h	未実装 「0」として 読み出し	620h	未実装 「0」として 読み出し	6A0h	未実装 「0」として 読み出し	720h	未実装 「0」として 読み出し	7A0h	未実装 「0」として 読み出し
46Fh	アクセス 70h ~ 7Fh	4EFh	アクセス 70h ~ 7Fh	56Fh	アクセス 70h ~ 7Fh	5EFh	アクセス 70h ~ 7Fh	66Fh	アクセス 70h ~ 7Fh	6EFh	アクセス 70h ~ 7Fh	76Fh	アクセス 70h ~ 7Fh	7EFh	アクセス 70h ~ 7Fh
470h	—	4F0h	—	570h	—	5F0h	—	670h	—	6F0h	—	770h	—	7F0h	—
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	コアレジスタ (表 3-2)	880h	コアレジスタ (表 3-2)	900h	コアレジスタ (表 3-2)	980h	コアレジスタ (表 3-2)	A00h	コアレジスタ (表 3-2)	A80h	コアレジスタ (表 3-2)	B00h	コアレジスタ (表 3-2)	B80h	コアレジスタ (表 3-2)
80Bh	未実装 「0」として 読み出し	88Bh	未実装 「0」として 読み出し	90Bh	未実装 「0」として 読み出し	98Bh	未実装 「0」として 読み出し	A0Bh	未実装 「0」として 読み出し	A8Bh	未実装 「0」として 読み出し	B0Bh	未実装 「0」として 読み出し	B8Bh	未実装 「0」として 読み出し
80Ch	—	88Ch	—	90Ch	—	98Ch	—	A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
86Fh	アクセス 70h ~ 7Fh	8EFh	アクセス 70h ~ 7Fh	96Fh	アクセス 70h ~ 7Fh	9EFh	アクセス 70h ~ 7Fh	A6Fh	アクセス 70h ~ 7Fh	A6Fh	アクセス 70h ~ 7Fh	B6Fh	アクセス 70h ~ 7Fh	BEFh	アクセス 70h ~ 7Fh
870h	—	8F0h	—	970h	—	9F0h	—	A70h	—	A70h	—	B70h	—	BF0h	—
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	A7Fh	—	B7Fh	—	BFh	—

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

表 3-8: PIC16(L)F1765 のメモリマップ (BANK 8 ~ 23)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	コアレジスタ (表 3-2)	480h	コアレジスタ (表 3-2)	500h	コアレジスタ (表 3-2)	580h	コアレジスタ (表 3-2)	600h	コアレジスタ (表 3-2)	680h	コアレジスタ (表 3-2)	700h	コアレジスタ (表 3-2)	780h	コアレジスタ (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACL	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	—	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	—	594h	—	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	—	595h	—	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	—	596h	—	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	—	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Ah	—	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	—
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	—	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	—	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	—
420h	汎用レジスタ 80 バイト	4A0h	汎用レジスタ 80 バイト	520h	汎用レジスタ 80 バイト	5A0h	汎用レジスタ 80 バイト	620h	汎用レジスタ 48 バイト	6A0h	未実装 「0」として 読み出し	720h	未実装 「0」として 読み出し	7A0h	未実装 「0」として 読み出し
46Fh	アクセス 70h ~ 7Fh	4EFh	アクセス 70h ~ 7Fh	56Fh	アクセス 70h ~ 7Fh	5EFh	アクセス 70h ~ 7Fh	66Fh	アクセス 70h ~ 7Fh	6EFh	アクセス 70h ~ 7Fh	76Fh	アクセス 70h ~ 7Fh	7EFh	アクセス 70h ~ 7Fh
470h	—	4F0h	—	570h	—	5F0h	—	670h	—	6F0h	—	770h	—	7F0h	—
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	コアレジスタ (表 3-2)	880h	コアレジスタ (表 3-2)	900h	コアレジスタ (表 3-2)	980h	コアレジスタ (表 3-2)	A00h	コアレジスタ (表 3-2)	A80h	コアレジスタ (表 3-2)	B00h	コアレジスタ (表 3-2)	B80h	コアレジスタ (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未実装 「0」として 読み出し	88Ch	未実装 「0」として 読み出し	90Ch	未実装 「0」として 読み出し	98Ch	未実装 「0」として 読み出し	A0Ch	未実装 「0」として 読み出し	A8Ch	未実装 「0」として 読み出し	B0Ch	未実装 「0」として 読み出し	B8Ch	未実装 「0」として 読み出し
86Fh	—	8EFh	—	96Fh	—	9EFh	—	A6Fh	—	AEFh	—	B6Fh	—	BEFh	—
870h	アクセス 70h ~ 7Fh	8F0h	アクセス 70h ~ 7Fh	970h	アクセス 70h ~ 7Fh	9F0h	アクセス 70h ~ 7Fh	A70h	アクセス 70h ~ 7Fh	AF0h	アクセス 70h ~ 7Fh	B70h	アクセス 70h ~ 7Fh	BF0h	アクセス 70h ~ 7Fh
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFh	—	B7Fh	—	BFh	—

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

表 3-9: PIC16(L)F1768 のメモリマップ (BANK 8 ~ 23)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	コアレジスタ (表 3-2)	480h	コアレジスタ (表 3-2)	500h	コアレジスタ (表 3-2)	580h	コアレジスタ (表 3-2)	600h	コアレジスタ (表 3-2)	680h	コアレジスタ (表 3-2)	700h	コアレジスタ (表 3-2)	780h	コアレジスタ (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACL	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	OPA2NCHS	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	OPA2PCHS	594h	DAC2CON0	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	OPA2CON	595h	DAC2REFL	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	OPA2ORS	596h	DAC2REFH	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	DAC4CON0	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	DAC4REF	61Ah	PWM4DCL	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	PRG2RTSS
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	PWM4DCH	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	PRG2FTSS
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	PRG2INS
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	PRG2CON0
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	PRG2CON1
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	PRG2CON2
420h	未実装 「0」として 読み出し	4A0h	未実装 「0」として 読み出し	520h	未実装 「0」として 読み出し	5A0h	未実装 「0」として 読み出し	620h	未実装 「0」として 読み出し	6A0h	未実装 「0」として 読み出し	720h	未実装 「0」として 読み出し	7A0h	未実装 「0」として 読み出し
46Fh	アクセス 70h ~ 7Fh	4EFh	アクセス 70h ~ 7Fh	56Fh	アクセス 70h ~ 7Fh	5EFh	アクセス 70h ~ 7Fh	66Fh	アクセス 70h ~ 7Fh	6EFh	アクセス 70h ~ 7Fh	76Fh	アクセス 70h ~ 7Fh	7EFh	アクセス 70h ~ 7Fh
470h	—	4F0h	—	570h	—	5F0h	—	670h	—	6F0h	—	770h	—	7F0h	—
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	コアレジスタ (表 3-2)	880h	コアレジスタ (表 3-2)	900h	コアレジスタ (表 3-2)	980h	コアレジスタ (表 3-2)	A00h	コアレジスタ (表 3-2)	A80h	コアレジスタ (表 3-2)	B00h	コアレジスタ (表 3-2)	B80h	コアレジスタ (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未実装 「0」として 読み出し	88Ch	未実装 「0」として 読み出し	90Ch	未実装 「0」として 読み出し	98Ch	未実装 「0」として 読み出し	A0Ch	未実装 「0」として 読み出し	A8Ch	未実装 「0」として 読み出し	B0Ch	未実装 「0」として 読み出し	B8Ch	未実装 「0」として 読み出し
86Fh	アクセス 70h ~ 7Fh	8EFh	アクセス 70h ~ 7Fh	96Fh	アクセス 70h ~ 7Fh	9EFh	アクセス 70h ~ 7Fh	A6Fh	アクセス 70h ~ 7Fh	A6Fh	アクセス 70h ~ 7Fh	B6Fh	アクセス 70h ~ 7Fh	BEFh	アクセス 70h ~ 7Fh
870h	—	8F0h	—	970h	—	9F0h	—	A70h	—	A70h	—	B70h	—	BF0h	—
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFFh	—	B7Fh	—	BFFh	—

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

表 3-10: PIC16(L)F1769 のメモリマップ (BANK 8 ~ 23)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	コアレジスタ (表 3-2)	480h	コアレジスタ (表 3-2)	500h	コアレジスタ (表 3-2)	580h	コアレジスタ (表 3-2)	600h	コアレジスタ (表 3-2)	680h	コアレジスタ (表 3-2)	700h	コアレジスタ (表 3-2)	780h	コアレジスタ (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACL	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	OPA2NCHS	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	OPA2PCHS	594h	DAC2CON0	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	OPA2CON	595h	DAC2REFL	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	OPA2ORS	596h	DAC2REFH	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	DAC4CON0	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	DAC4REF	61Ah	PWM4DCL	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	PRG2RTSS
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	PWM4DCH	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	PRG2FTSS
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	PRG2INS
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	PRG2CON0
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	PRG2CON1
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	PRG2CON2
420h	汎用レジスタ 80 バイト	4A0h	汎用レジスタ 80 バイト	520h	汎用レジスタ 80 バイト	5A0h	汎用レジスタ 80 バイト	620h	汎用レジスタ 48 バイト	6A0h	未実装 「0」として 読み出し	720h	未実装 「0」として 読み出し	7A0h	未実装 「0」として 読み出し
46Fh	アクセス 70h ~ 7Fh	4EFh	アクセス 70h ~ 7Fh	56Fh	アクセス 70h ~ 7Fh	5EFh	アクセス 70h ~ 7Fh	66Fh	アクセス 70h ~ 7Fh	6EFh	アクセス 70h ~ 7Fh	76Fh	アクセス 70h ~ 7Fh	7EFh	アクセス 70h ~ 7Fh
470h	—	4F0h	—	570h	—	5F0h	—	670h	—	6F0h	—	770h	—	7F0h	—
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	コアレジスタ (表 3-2)	880h	コアレジスタ (表 3-2)	900h	コアレジスタ (表 3-2)	980h	コアレジスタ (表 3-2)	A00h	コアレジスタ (表 3-2)	A80h	コアレジスタ (表 3-2)	B00h	コアレジスタ (表 3-2)	B80h	コアレジスタ (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未実装 「0」として 読み出し	88Ch	未実装 「0」として 読み出し	90Ch	未実装 「0」として 読み出し	98Ch	未実装 「0」として 読み出し	A0Ch	未実装 「0」として 読み出し	A8Ch	未実装 「0」として 読み出し	B0Ch	未実装 「0」として 読み出し	B8Ch	未実装 「0」として 読み出し
86Fh	—	8EFh	—	96Fh	—	9EFh	—	A6Fh	—	AEFh	—	B6Fh	—	BEFh	—
870h	アクセス 70h ~ 7Fh	8F0h	アクセス 70h ~ 7Fh	970h	アクセス 70h ~ 7Fh	9F0h	アクセス 70h ~ 7Fh	A70h	アクセス 70h ~ 7Fh	AF0h	アクセス 70h ~ 7Fh	B70h	アクセス 70h ~ 7Fh	BF0h	アクセス 70h ~ 7Fh
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFh	—	B7Fh	—	BFh	—

凡例: ■ = 未実装のデータメモリ領域、「0」として読み出し

表 3-11: PIC16(L)F1764/5/8/9 のメモリマップ (BANK 24 ~ 31)

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	コアレジスタ (表 3-2)	C80h	コアレジスタ (表 3-2)	D00h	コアレジスタ (表 3-2)	D80h	コアレジスタ (表 3-2)	E00h	コアレジスタ (表 3-2)	E80h	コアレジスタ (表 3-2)	F00h	コアレジスタ (表 3-2)	F80h	コアレジスタ (表 3-2)
C0Bh	—	C8Bh	—	D0Bh	—	D8Bh	詳細は 表 3-12 と表 3-13 参照	E0Bh	詳細は 表 3-12 と表 3-13 参照	E8Bh	詳細は 表 3-12 と表 3-13 参照	F0Bh	詳細は 表 3-12 と表 3-13 参照	F8Bh	詳細は 表 3-14 参照
C0Ch	—	C8Ch	—	D0Ch	—										
C0Dh	—	C8Dh	—	D0Dh	—										
C0Eh	—	C8Eh	—	D0Eh	—										
C0Fh	—	C8Fh	—	D0Fh	—										
C10h	—	C90h	—	D10h	—										
C11h	—	C91h	—	D11h	—										
C12h	—	C92h	—	D12h	—										
C13h	—	C93h	—	D13h	—										
C14h	—	C94h	—	D14h	—										
C15h	—	C95h	—	D15h	—										
C16h	—	C96h	—	D16h	—										
C17h	—	C97h	—	D17h	—										
C18h	—	C98h	—	D18h	—										
C19h	—	C99h	—	D19h	—										
C1Ah	—	C9Ah	—	D1Ah	—										
C1Bh	—	C9Bh	—	D1Bh	—										
C1Ch	—	C9Ch	—	D1Ch	—										
C1Dh	—	C9Dh	—	D1Dh	—										
C1Eh	—	C9Eh	—	D1Eh	—										
C1Fh	—	C9Fh	—	D1Fh	—										
C20h	—	CA0h	—	D20h	—										
C6Fh	未実装 「0」として 読み出し	CEFh	未実装 「0」として 読み出し	D6Fh	未実装 「0」として 読み出し	DEFh	アクセス 70h ~ 7Fh	E6Fh	アクセス 70h ~ 7Fh	EEFh	アクセス 70h ~ 7Fh	F6Fh	アクセス 70h ~ 7Fh	FEFh	アクセス 70h ~ 7Fh
C70h	アクセス 70h ~ 7Fh	CF0h	アクセス 70h ~ 7Fh	D70h	アクセス 70h ~ 7Fh	DF0h		E70h		EF0h		F70h		FF0h	
CFh	—	CFh	—	D7Fh	—	DFh	—	E7Fh	—	EFh	—	F7Fh	—	FFh	—

凡例: = 未実装のデータメモリ領域、「0」として読み出し

表 3-12: PIC16(L)F1764/5 のメモリマップ (BANK 27 ~ 30)

BANK 27		BANK 28		BANK 29		BANK 30	
D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—
D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—
D8Eh	PWMEN	E0Eh	—	E8Eh	—	F0Eh	—
D8Fh	PWMLD	E0Fh	PPSLOCK	E8Fh	—	F0Fh	CLCDATA
D90h	PWMOUT	E10h	INTPPS	E90h	RA0PPS	F10h	CLC1CON
D91h	PWM5PHL	E11h	T0CKIPPS	E91h	RA1PPS	F11h	CLC1POL
D92h	PWM5PHH	E12h	T1CKIPPS	E92h	RA2PPS	F12h	CLC1SEL0
D93h	PWM5DCL	E13h	T1GPPS	E93h	—	F13h	CLC1SEL1
D94h	PWM5DCH	E14h	CCP1PPS	E94h	RA4PPS	F14h	CLC1SEL2
D95h	PWM5PRL	E15h	—	E95h	RA5PPS	F15h	CLC1SEL3
D96h	PWM5PRH	E16h	COG1INPPS	E96h	—	F16h	CLC1GLS0
D97h	PWM5OFL	E17h	—	E97h	—	F17h	CLC1GLS1
D98h	PWM5OFH	E18h	—	E98h	—	F18h	CLC1GLS2
D99h	PWM5TMRL	E19h	T2CKIPPS	E99h	—	F19h	CLC1GLS3
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS	E9Ah	—	F1Ah	CLC2CON
D9Bh	PWM5CON	E1Bh	T3GPPS	E9Bh	—	F1Bh	CLC2POL
D9Ch	PWM5INTE	E1Ch	T4CKIPPS	E9Ch	—	F1Ch	CLC2SEL0
D9Dh	PWM5INTF	E1Dh	T5CKIPPS	E9Dh	—	F1Dh	CLC2SEL1
D9Eh	PWM5CLKCON	E1Eh	T5GPPS	E9Eh	—	F1Eh	CLC2SEL2
D9Fh	PWM5LDCON	E1Fh	T6CKIPPS	E9Fh	—	F1Fh	CLC2SEL3
DA0h	PWM5OFCON	E20h	SSPCKLPPS	EA0h	RC0PPS	F20h	CLC2GLS0
DA1h	—	E21h	SSPDATPPS	EA1h	RC1PPS	F21h	CLC2GLS1
DA2h	—	E22h	SSPSSPPS	EA2h	RC2PPS	F22h	CLC2GLS2
DA3h	—	E23h	—	EA3h	RC3PPS	F23h	CLC2GLS3
DA4h	—	E24h	RXPPS	EA4h	RC4PPS	F24h	CLC3CON
DA5h	—	E25h	CKPPS	EA5h	RC5PPS	F25h	CLC3POL
DA6h	—	E26h	—	EA6h	—	F26h	CLC3SEL0
DA7h	—	E27h	—	EA7h	—	F27h	CLC3SEL1
DA8h	—	E28h	CLCIN0PPS	EA8h	—	F28h	CLC3SEL2
DA9h	—	E29h	CLCIN1PPS	EA9h	—	F29h	CLC3SEL3
DAAh	—	E2Ah	CLCIN2PPS	EAAh	—	F2Ah	CLC3GLS0
DABh	—	E2Bh	CLCIN3PPS	EABh	—	F2Bh	CLC3GLS1
DACH	—	E2Ch	PRG1FPPS	EACH	—	F2Ch	CLC3GLS2
DADh	—	E2Dh	PRG1RPPS	EADh	—	F2Dh	CLC3GLS3
DAEh	—	E2Eh	—	EA Eh	—	F2Eh	—
DAFh	—	E2Fh	—	EAFh	—	F2Fh	—
DB0h	—	E30h	MD1CHPPS	EB0h	—	F30h	—
DB1h	—	E31h	MD1CLPPS	EB1h	—	F31h	—
DB2h	—	E32h	MD1MODPPS	EB2h	—	F32h	—
DB3h	—	E33h	—	EB3h	—	F33h	—
DB4h	—	E34h	—	EB4h	—	F34h	—
DB5h	—	E35h	—	EB5h	—	F35h	—
DB6h	—	E36h	—	EB6h	—	F36h	—
DB7h	—	E37h	—	EB7h	—	F37h	—
DB8h	—	E38h	—	EB8h	—	F38h	—
DB9h	—	E39h	—	EB9h	—	F39h	—
DBAh	—	E3Ah	—	EBAh	—	F3Ah	—
DBBh	—	E3Bh	—	EBBh	—	F3Bh	—
DBCh	—	E3Ch	—	EBCh	—	F3Ch	—
DBDh	—	E3Dh	—	EBDh	—	F3Dh	—
DBEh	—	E3Eh	—	EBEh	—	F3Eh	—
DBFh	—	E3Fh	—	EBFh	—	F3Fh	—
DC0h	—	E40h	—	EC0h	—	F40h	—
DEFh	—	E6Fh	—	EEFh	—	F6Fh	—

凡例: = 未実装のデータメモリ領域、「0」として読み出し

PIC16(L)F1764/5/8/9

表 3-13: PIC16(L)F1768/9 のメモリマップ (BANK 27 ~ 30)

BANK 27			BANK 28			BANK 29			BANK 30		
D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—				
D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—				
D8Eh	PWMEN	E0Eh	—	E8Eh	—	F0Eh	—				
D8Fh	PWMLD	E0Fh	PPSLOCK	E8Fh	—	F0Fh	CLCDATA				
D90h	PWMOUT	E10h	INTPPS	E90h	RA0PPS	F10h	CLC1CON				
D91h	PWM5PHL	E11h	T0CKIPPS	E91h	RA1PPS	F11h	CLC1POL				
D92h	PWM5PHH	E12h	T1CKIPPS	E92h	RA2PPS	F12h	CLC1SEL0				
D93h	PWM5DCL	E13h	T1GPPS	E93h	—	F13h	CLC1SEL1				
D94h	PWM5DCH	E14h	CCP1PPS	E94h	RA4PPS	F14h	CLC1SEL2				
D95h	PWM5PRL	E15h	CCP2PPS	E95h	RA5PPS	F15h	CLC1SEL3				
D96h	PWM5PRH	E16h	COG1INPPS	E96h	—	F16h	CLC1GLS0				
D97h	PWM5OFL	E17h	COG2INPPS	E97h	—	F17h	CLC1GLS1				
D98h	PWM5OFH	E18h	—	E98h	—	F18h	CLC1GLS2				
D99h	PWM5TMRL	E19h	T2CKIPPS	E99h	—	F19h	CLC1GLS3				
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS	E9Ah	—	F1Ah	CLC2CON				
D9Bh	PWM5CON	E1Bh	T3GPPS	E9Bh	—	F1Bh	CLC2POL				
D9Ch	PWM5INTE	E1Ch	T4CKIPPS	E9Ch	RB4PPS	F1Ch	CLC2SEL0				
D9Dh	PWM5INTF	E1Dh	T5CKIPPS	E9Dh	RB5PPS	F1Dh	CLC2SEL1				
D9Eh	PWM5CLKCON	E1Eh	T5GPPS	E9Eh	RB6PPS	F1Eh	CLC2SEL2				
D9Fh	PWM5LDCON	E1Fh	T6CKIPPS	E9Fh	RB7PPS	F1Fh	CLC2SEL3				
DA0h	PWM5OFCON	E20h	SSPCLKPPS	EA0h	RC0PPS	F20h	CLC2GLS0				
DA1h	PWM6PHL	E21h	SSPDATPPS	EA1h	RC1PPS	F21h	CLC2GLS1				
DA2h	PWM6PHH	E22h	SSPSSPPS	EA2h	RC2PPS	F22h	CLC2GLS2				
DA3h	PWM6DCL	E23h	—	EA3h	RC3PPS	F23h	CLC2GLS3				
DA4h	PWM6DCH	E24h	RXPPS	EA4h	RC4PPS	F24h	CLC3CON				
DA5h	PWM6PRL	E25h	CKPPS	EA5h	RC5PPS	F25h	CLC3POL				
DA6h	PWM6PRH	E26h	—	EA6h	RC6PPS	F26h	CLC3SEL0				
DA7h	PWM6OFL	E27h	—	EA7h	RC7PPS	F27h	CLC3SEL1				
DA8h	PWM6OFH	E28h	CLCIN0PPS	EA8h	—	F28h	CLC3SEL2				
DA9h	PWM6TMRL	E29h	CLCIN1PPS	EA9h	—	F29h	CLC3SEL3				
DAAh	PWM6TMRH	E2Ah	CLCIN2PPS	EAAh	—	F2Ah	CLC3GLS0				
DABh	PWM6CON	E2Bh	CLCIN3PPS	EABh	—	F2Bh	CLC3GLS1				
DACh	PWM6INTE	E2Ch	PRG1FPPS	EACH	—	F2Ch	CLC3GLS2				
DADh	PWM6INTF	E2Dh	PRG1RPPS	EADh	—	F2Dh	CLC3GLS3				
DAEh	PWM6CLKCON	E2Eh	PRG2FPPS	EAeh	—	F2Eh	—				
DAFh	PWM6LDCON	E2Fh	PRG2RPPS	EAfh	—	F2Fh	—				
DB0h	PWM6OFCON	E30h	MD1CHPPS	EB0h	—	F30h	—				
DB1h	—	E31h	MD1CLPPS	EB1h	—	F31h	—				
DB2h	—	E32h	MD1MODPPS	EB2h	—	F32h	—				
DB3h	—	E33h	MD2CHPPS	EB3h	—	F33h	—				
DB4h	—	E34h	MD2CLPPS	EB4h	—	F34h	—				
DB5h	—	E35h	MD2MODPPS	EB5h	—	F35h	—				
DB6h	—	E36h	—	EB6h	—	F36h	—				
DB7h	—	E37h	—	EB7h	—	F37h	—				
DB8h	—	E38h	—	EB8h	—	F38h	—				
DB9h	—	E39h	—	EB9h	—	F39h	—				
DBAh	—	E3Ah	—	EBAh	—	F3Ah	—				
DBBh	—	E3Bh	—	EBBh	—	F3Bh	—				
DBCh	—	E3Ch	—	EBCh	—	F3Ch	—				
DBDh	—	E3Dh	—	EBDh	—	F3Dh	—				
DBEh	—	E3Eh	—	EBEh	—	F3Eh	—				
DBFh	—	E3Fh	—	EBFh	—	F3Fh	—				
DC0h	—	E40h	—	EC0h	—	F40h	—				
DEFh	—	E6Fh	—	EEFh	—	F6Fh	—				

凡例: = 未実装のデータメモリ領域、「0」として読み出し

表 3-14: PIC16(L)F1764/5/8/9 のメモリマップ
(BANK 31)

BANK 31	
F8Ch	未実装 「0」として読み出し
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

凡例: = 未実装のデータメモリ領域、
「0」として読み出し

PIC16(L)F1764/5/8/9

3.4.5 コア機能レジスタのまとめ

表 3-15 に一覧を示したコア機能レジスタは、任意のバンクからアドレス指定できます。

表 3-15: コア機能レジスタのまとめ (1)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他の リセット時 の値	
BANK 0 ~ 31												
x00h または x80h	INDF0	この位置をアドレス指定すると、FSR0H/FSR0L の内容によってデータメモリのアドレスを指定する (物理レジスタではない)。								xxxx xxxx	uuuu uuuu	
x01h または x81h	INDF1	この位置をアドレス指定すると、FSR1H/FSR1L の内容によってデータメモリのアドレスを指定する (物理レジスタではない)。								xxxx xxxx	uuuu uuuu	
x02h または x82h	PCL	プログラムカウンタ (PC) の下位バイト								0000 0000	0000 0000	
x03h または x83h	STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu	
x04h または x84h	FSR0L	間接データメモリ アドレスポインタ 0、下位バイト								0000 0000	uuuu uuuu	
x05h または x85h	FSR0H	間接データメモリ アドレスポインタ 0、上位バイト								0000 0000	0000 0000	
x06h または x86h	FSR1L	間接データメモリ アドレスポインタ 1、下位バイト								0000 0000	uuuu uuuu	
x07h または x87h	FSR1H	間接データメモリ アドレスポインタ 1、上位バイト								0000 0000	0000 0000	
x08h または x88h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
x09h または x89h	WREG	ワーキング レジスタ								0000 0000	uuuu uuuu	
x0Ah または x8Ah	PCLATH	—	プログラムカウンタ上位 7 ビットの書き込みバッファ								-000 0000	-000 0000
x0Bh または x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

Note 1: これらのレジスタへは、全バンクからアクセスできます。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK 0												
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu	
00Dh	PORTB ⁽²⁾	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----	
00Eh	PORTC	RC7 ⁽²⁾	RC6 ⁽²⁾	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu	
00Fh	—	未実装									—	—
010h	—	未実装									—	—
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
012h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽²⁾	C3IF ⁽²⁾	CCP2IF ⁽²⁾	000- 0000	000- 0000	
013h	PIR3	PWM6IF ⁽²⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽²⁾	CLC3IF	CLC2IF	CLC1IF	0000 0000	0000 0000	
014h	PIR4	—	—	TMR5GIF	TMR5IF	TMR3GIF	TMR3IF	TMR6IF	TRM4IF	--00 0000	--00 0000	
015h	TMR0	Timer0 モジュール レジスタ									0000 0000	0000 0000
016h	TMR1L	16 ビット TMR1 レジスタの下位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
017h	TMR1H	16 ビット TMR1 レジスタの上位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
018h	T1CON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON	0000 00-0	uuuu uu-u	
019h	T1GCON	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu	
01Ah	T2TMR	8 ビット TMR2 レジスタ用の保持レジスタ									0000 0000	0000 0000
01Bh	T2PR	TMR2 周期レジスタ									1111 1111	1111 1111
01Ch	T2CON	ON	CKPS<2:0>			OUTPS<3:0>			0000 0000			0000 0000
01Dh	T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>						---0 0000	---0 0000
01Eh	T2CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000	
01Fh	T2RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000	
BANK 1												
08Ch	TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111	
08Dh	TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----	
08Eh	TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111	
08Fh	—	未実装									—	—
090h	—	未実装									—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
092h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽²⁾	C3IE ⁽²⁾	CCP2IE ⁽²⁾	000- 0000	000- 0000	
093h	PIE3	PWM6IE ⁽²⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽²⁾	CLC3IE	CLC2IE	CLC1IE	0000 0000	0000 0000	
094h	PIE4	—	—	TMR5GIE	TMR5IE	TMR3GIE	TMR3IE	TMR6IE	TRM4IE	--00 0000	--00 0000	
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111	
096h	PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	00-1 11qq	qq-q qquu	
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110	
098h	OSCTUNE	—	—	TUN<5:0>						--00 0000	--00 0000	
099h	OSCCON	SPLLEN	IRCF<3:0>			—	SCS<1:0>			0011 1-00	0011 1-00	
09Ah	OSCSTAT	SOSCR	PLL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	00q0 0q0q	qqqq qq0q	
09Bh	ADRESL	A/D 変換結果レジスタ下位									xxxx xxxx	uuuu uuuu
09Ch	ADRESH	A/D 変換結果レジスタ上位									xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/DONE	ADON	-000 0000	-000 0000	
09Eh	ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>			0000 -000	0000 -000
09Fh	ADCON2	TRIGSEL<4:0>									0000 0---	0000 0---

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値		
BANK 2													
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu		
10Dh	LATB ⁽²⁾	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxx ----	uuuu ----		
10Eh	LATC	LATC7 ⁽²⁾	LATC6 ⁽²⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu		
10Fh	CMOUT	—	—	—	—	MC4OUT ⁽²⁾	MC3OUT ⁽²⁾	MC2OUT	MC1OUT	---- --00	---- --00		
110h	CM1CON0	ON	OUT	—	POL	ZLF	予約済み	HYS	SYNC	00-0 0100	00-0 0100		
111h	CM1CON1	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00		
112h	CM1NSEL	—	—	—	—	—	NCH<2:0>			---- -000	---- -000		
113h	CM1PSEL	—	—	—	—	PCH<2:0>				---- 0000	---- 0000		
114h	CM2CON0	ON	OUT	—	POL	ZLF	予約済み	HYS	SYNC	00-0 0100	00-0 0100		
115h	CM2CON1	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00		
116h	CM2NSEL	—	—	—	—	—	NCH<2:0>			---- -000	---- -000		
117h	CM2PSEL	—	—	—	—	PCH<2:0>				---- 0000	---- 0000		
118h	CM3CON0 ⁽²⁾	ON	OUT	—	POL	ZLF	予約済み	HYS	SYNC	00-0 0100	00-0 0100		
119h	CM3CON1 ⁽²⁾	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00		
11Ah	CM3NSEL ⁽²⁾	—	—	—	—	—	NCH<2:0>			---- -000	---- -000		
11Bh	CM3PSEL ⁽²⁾	—	—	—	—	PCH<2:0>				---- 0000	---- 0000		
11Ch	CM4CON0 ⁽²⁾	ON	OUT	—	POL	ZLF	予約済み	HYS	SYNC	00-0 0100	00-0 0100		
11Dh	CM4CON1 ⁽²⁾	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00		
11Eh	CM4NSEL ⁽²⁾	—	—	—	—	—	NCH<2:0>			---- -000	---- -000		
11Fh	CM4PSEL ⁽²⁾	—	—	—	—	PCH<2:0>				---- 0000	---- 0000		
BANK 3													
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 1111	---1 1111		
18Dh	ANSELB ⁽²⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	1111 ----	1111 ----		
18Eh	ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111		
18Fh	—	未実装									—	—	
190h	—	未実装									—	—	
191h	PMADRL	プログラムメモリ アドレスレジスタ下位バイト									0000 0000	0000 0000	
192h	PMADRH	— ⁽¹⁾	プログラムメモリ アドレスレジスタ上位バイト									1000 0000	1000 0000
193h	PMDATL	プログラムメモリ読み出しデータレジスタ下位バイト									xxxx xxxx	uuuu uuuu	
194h	PMDATH	—	—	プログラムメモリ読み出しデータレジスタ上位バイト								--xx xxxx	--uu uuuu
195h	PMCON1	— ⁽¹⁾	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000		
196h	PMCON2	プログラムメモリ制御レジスタ 2									0000 0000	0000 0000	
197h	VREGCON ⁽⁴⁾	—	—	—	—	—	—	VREGPM	予約済み	---- --01	---- --01		
198h	—	未実装									—	—	
199h	RC1REG	EUSART 受信データレジスタ									0000 0000	0000 0000	
19Ah	TX1REG	EUSART 送信データレジスタ									0000 0000	0000 0000	
19Bh	SP1BRGL	SP1BRG<7:0>									0000 0000	0000 0000	
19Ch	SP1BRGH	SP1BRG<15:8>									0000 0000	0000 0000	
19Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 0000	0000 0000		
19Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010		
19Fh	BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00		

凡例: x = 未知, u = 不変, q = 条件による, - = 未実装, 「0」として読み出し, r = 予約済み
網掛け表示されている位置は未実装, 「0」として読み出し

- Note
- 1: 未実装, 「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK 4												
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111	
20Dh	WPUB ⁽²⁾	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----	
20Eh	WPUC	WPUC7 ⁽²⁾	WPUC6 ⁽²⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111	1111 1111	
20Fh	—	未実装									—	—
210h	—	未実装									—	—
211h	SSP1BUF	同期シリアルポート受信バッファ / 送信レジスタ								xxxx xxxx	uuuu uuuu	
212h	SSP1ADD	ADD<7:0>								0000 0000	0000 0000	
213h	SSP1MSK	MSK<7:0>								1111 1111	1111 1111	
214h	SSP1STAT	SMP	CKE	D/Ā	P	S	R \bar{W}	UA	BF	0000 0000	0000 0000	
215h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000	
216h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000	
217h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000	
218h — 21Ch	—	未実装									—	—
21Dh	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u	
21Eh	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000	0q00 0000	
21Fh	ZCD1CON	EN	—	OUT	POL	—	—	INTP	INTN	0-x0 --00	0-x0 --00	
BANK 5												
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	--00 -000	--00 -000	
28Dh	ODCONB ⁽²⁾	ODB7	ODB6	ODB5	ODB4	—	—	—	—	0000 ----	0000 ----	
28Eh	ODCONC	ODC7 ⁽²⁾	ODC6 ⁽²⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000 0000	0000 0000	
28Fh	—	未実装									—	—
290h	—	未実装									—	—
291h	CCPR1L	キャプチャ / コンペア / PWM レジスタ 1 (LSB)								xxxx xxxx	uuuu uuuu	
292h	CCPR1H	キャプチャ / コンペア / PWM レジスタ 1 (MSB)								xxxx xxxx	uuuu uuuu	
293h	CCP1CON	EN	—	OUT	FMT	MODE<3:0>			0-00 0000	0-00 0000		
294h	CCP1CAP	—	—	—	—	—	CTS<2:0>			---- -000	---- -000	
295h — 297h	—	未実装									—	—
298h	CCPR2L ⁽²⁾	キャプチャ / コンペア / PWM レジスタ 2 (LSB)								xxxx xxxx	uuuu uuuu	
299h	CCPR2H ⁽²⁾	キャプチャ / コンペア / PWM レジスタ 2 (MSB)								xxxx xxxx	uuuu uuuu	
29Ah	CCP2CON ⁽²⁾	EN	—	OUT	FMT	MODE<3:0>			0-00 0000	0-00 0000		
29Bh	CCP2CAP ⁽²⁾	—	—	—	—	—	CTS<2:0>			---- -000	---- -000	
29Ch — 29Dh	—	未実装									—	—
29Eh	CCPTMRS	P4TSEL<1:0> ⁽²⁾		P3TSEL<1:0>		C2TSEL<1:0> ⁽²⁾		C1TSEL<1:0>		0000 0000	0000 0000	
29Fh	—	未実装									—	—

凡例: ※ = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、x = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note 1: 未実装、「1」として読み出します。
2: PIC16(L)F1768/9 のみです。
3: PIC16(L)F1764/5 のみです。
4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値
BANK 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 -111	--11 -111
30Dh	SLRCONB ⁽²⁾	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----
30Eh	SLRCONC	SLRC7 ⁽²⁾	SLRC6 ⁽²⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
30Fh — 31Ah	—	未実装								—	—
31Bh	MD2CON0 ⁽²⁾	EN	—	OUT	OPOL	—	—	—	BIT	0-00 ---0	0-00 ---0
31Ch	MD2CON1 ⁽²⁾	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	--00 --00	--00 --00
31Dh	MD2SRC ⁽²⁾	—	—	—	MS<4:0>				—	--0 0000	--0 0000
31Eh	MD2CARL ⁽²⁾	—	—	—	—	CL<3:0>				---- 0000	---- 0000
31Fh	MD2CARH ⁽²⁾	—	—	—	—	CH<3:0>				---- 0000	---- 0000
BANK 7											
38Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
38Dh	INLVLB ⁽²⁾	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	INLVLC7 ⁽²⁾	INLVLC6 ⁽²⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
38Fh	—	未実装								—	—
390h	—	未実装								—	—
391h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
392h	IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
393h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
394h	IOCBP ⁽²⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN ⁽²⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF ⁽²⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h	IOCCP	IOCCP7 ⁽²⁾	IOCCP6 ⁽²⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	IOCCN7 ⁽²⁾	IOCCN6 ⁽²⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	IOCCF7 ⁽²⁾	IOCCF6 ⁽²⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
39Ah	—	未実装								—	—
39Bh	MD1CON0	EN	—	OUT	OPOL	—	—	—	BIT	0-00 ---0	0-00 ---0
39Ch	MD1CON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	--00 --00	--00 --00
39Dh	MD1SRC	—	—	—	MS<4:0>				—	--0 0000	--0 0000
39Eh	MD1CARL	—	—	—	—	CL<3:0>				---- 0000	---- 0000
39Fh	MD1CARH	—	—	—	—	CH<3:0>				---- 0000	---- 0000

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK 8												
40Ch ~ 40Dh	—	未実装								—	—	
40Eh	HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	--00 ----	--00 ----	
40Fh ~ 412h	—	未実装								—	—	
413h	T4TMR	8 ビット TMR4 レジスタ用の保持レジスタ									0000 0000	0000 0000
413h	T4PR	TMR4 周期レジスタ									1111 1111	1111 1111
415h	T4CON	ON	CKPS<2:0>			OUTPS<3:0>					0000 0000	0000 0000
416h	T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					---- 0000	---- 0000	
417h	T4CLKCON	—	—	—	—	CS<3:0>					---- 0000	---- 0000
418h	T4RST	—	—	—	—	RSEL<3:0>					---- 0000	---- 0000
419h	—	未実装								—	—	
41Ah	T6TMR	8 ビット TMR4 レジスタ用の保持レジスタ									0000 0000	0000 0000
41Bh	T6PR	TMR4 周期レジスタ									1111 1111	1111 1111
41Ch	T6CON	ON	CKPS<2:0>			OUTPS<3:0>					0000 0000	0000 0000
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					---- 0000	---- 0000	
41Eh	T6CLKCON	—	—	—	—	CS<3:0>					---- 0000	---- 0000
41Fh	T6RST	—	—	—	—	RSEL<3:0>					---- 0000	---- 0000
BANK 9												
48Ch ~ 492h	—	未実装								—	—	
493h	TMR3L	16 ビット TMR1 レジスタの下位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
494h	TMR3H	16 ビット TMR1 レジスタの上位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
495h	T3CON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON	0000 00-0	uuuu uu-u	
496h	T3GCON	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu	
497h ~ 499h	—	未実装								—	—	
49Ah	TMR5L	16 ビット TMR1 レジスタの下位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
49Bh	TMR5H	16 ビット TMR1 レジスタの上位バイト保持レジスタ									xxxx xxxx	uuuu uuuu
49Ch	T5CON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON	0000 00-0	uuuu uu-u	
49Dh	T5GCON	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu	
49Eh ~ 49Fh	—	未実装								—	—	

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note 1: 未実装、「1」として読み出します。
2: PIC16(L)F1768/9 のみです。
3: PIC16(L)F1764/5 のみです。
4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK 10												
50Ch — 50Eh	—	未実装								—	—	
50Fh	OPA1NCHS	—	—	—	—	NCH<3:0>				---- 0000	---- 0000	
510h	OPA1PCHS	—	—	—	—	PCH<3:0>				---- 0000	---- 0000	
511h	OPA1CON	EN	—	—	UG	—	ORPOL	ORM<1:0>		0--0 -000	0--0 -000	
512h	OPA1ORS	—	—	—	ORS<4:0>				---	0000	---	0000
513h	OPA2NCHS ⁽²⁾	—	—	—	—	NCH<3:0>				---- 0000	---- 0000	
514h	OPA2PCHS ⁽²⁾	—	—	—	—	PCH<3:0>				---- 0000	---- 0000	
515h	OPA2CON ⁽²⁾	EN	—	—	UG	—	ORPOL	ORM<1:0>		0--0 -000	0--0 -000	
516h	OPA2ORS ⁽²⁾	—	—	—	ORS<4:0>				---	0000	---	0000
517h — 51Fh	—	未実装								—	—	
BANK 11												
590h	DACL D	---	---	---	---	---	---	DAC2LD ⁽²⁾	DAC1LD	---- --00	---- --00	
591h	DAC1CON0	EN	FM	OE1	---	PSS<1:0>		---	NSS<1:0>	000- 0000	000- 0000	
592h	DAC1REFL	REF<7:0>								00000 0000	0000 0000	
593h	DAC1REFH	REF<15:8>								00000 0000	0000 0000	
594h	DAC2CON0 ⁽²⁾	EN	FM	OE1	---	PSS<1:0>		---	NSS<1:0>	000- 0000	000- 0000	
595h	DAC2REFL ⁽²⁾	REF<7:0>								00000 0000	0000 0000	
596h	DAC2REFH ⁽²⁾	REF<15:8>								00000 0000	0000 0000	
597h	DAC3CON0	EN	---	OE1	---	PSS<1:0>		---	NSS	0-0- 00-0	0-00 00-0	
598h	DAC3REF	---	---	---	REF<4:0>				---	0000	0000 0000	
599h	DAC4CON0 ⁽²⁾	EN	---	OE1	---	PSS<1:0>		---	NSS	0-0- 00-0	0-00 00-0	
59Ah	DAC4REF ⁽²⁾	---	---	---	REF<4:0>				---	0000	0000 0000	
59Bh ~ 59Fh	—	未実装								—	—	

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値
BANK 12											
60Ch ~ 616h	—	未実装								—	—
617h	PWM3DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
618h	PWM3DCH	DC<9:2>								xxxx xxxx	uuuu uuuu
619h	PWM3CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM4DCL ⁽²⁾	DC<1:0>		—	—	—	—	—	—	00-- ----	uu-- ----
61Bh	PWM4DCH ⁽²⁾	DC<9:2>								0000 0000	uuuu uuuu
61Ch	PWM4CON ⁽²⁾	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Dh — 61Fh	—	未実装								—	—
BANK 13											
68Ch	—	未実装								—	—
68Dh	COG1PHR	—	—	COG 立ち上がりエッジ位相遅延カウントレジスタ						--00 0000	--00 0000
68Eh	COG1PHF	—	—	COG 立ち下がりエッジ位相遅延カウントレジスタ						--00 0000	--00 0000
68Fh	COG1BLKR	—	—	COG 立ち上がりエッジブランキング カウントレジスタ						--00 0000	--00 0000
690h	COG1BLKF	—	—	COG 立ち下がりエッジブランキング カウントレジスタ						--00 0000	--00 0000
691h	COG1DBR	—	—	COG 立ち上がりエッジデッドバンド カウントレジスタ						--00 0000	--00 0000
692h	COG1DBF	—	—	COG 立ち下がりエッジデッドバンド カウントレジスタ						--00 0000	--00 0000
693h	COG1CON0	EN	LD	—	CS<1:0>		MD<2:0>			00-0 0000	00-0 0000
694h	COG1CON1	RDBS	FDDBS	—	—	POLD	POLC	POLB	POLA	00-- 0000	00-- 0000
695h	COG1RIS0	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	0000 0000	-000 0000
696h	COG1RIS1	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	--00 0000	-000 0000
697h	COG1RSIM0	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	0000 0000	-000 0000
698h	COG1RSIM1	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	--00 0000	-000 0000
699h	COG1FIS0	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	0000 0000	-000 0000
69Ah	COG1FIS1	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	--00 0000	-000 0000
69Bh	COG1FSIM0	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	0000 0000	-000 0000
69Ch	COG1FSIM1	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	--00 0000	-000 0000
69Dh	COG1ASD0	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—	0001 01--	0001 01--
69Eh	COG1ASD1	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	0000 0000	0000 0000
69Fh	COG1STR	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	0000 0000	0000 0000

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値		
BANK 14													
70Ch	—	未実装									—	—	
70Dh	COG2PHR ⁽²⁾	—	—	COG 立ち上がりエッジ位相遅延カウントレジスタ								--00 0000	--00 0000
70Eh	COG2PHF ⁽²⁾	—	—	COG 立ち下がりエッジ位相遅延カウントレジスタ								--00 0000	--00 0000
70Fh	COG2BLKR ⁽²⁾	—	—	COG 立ち上がりエッジ ブランキング カウントレジスタ								--00 0000	--00 0000
710h	COG2BLKF ⁽²⁾	—	—	COG 立ち下がりエッジ ブランキング カウントレジスタ								--00 0000	--00 0000
711h	COG2DBR ⁽²⁾	—	—	COG 立ち上がりエッジ デッドバンド カウントレジスタ								--00 0000	--00 0000
712h	COG2DBF ⁽²⁾	—	—	COG 立ち下がりエッジ デッドバンド カウントレジスタ								--00 0000	--00 0000
713h	COG2CON0 ⁽²⁾	EN	LD	—	CS<1:0>			MD<2:0>			00-0 0000	00-0 0000	
714h	COG2CON1 ⁽²⁾	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	00-- 0000	00-- 0000		
715h	COG2RIS0 ⁽²⁾	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	0000 0000	-000 0000		
716h	COG2RIS1 ⁽²⁾	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	--00 0000	-000 0000		
717h	COG2RSIM0 ⁽²⁾	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	0000 0000	-000 0000		
718h	COG2RSIM1 ⁽²⁾	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	--00 0000	-000 0000		
719h	COG2FIS0 ⁽²⁾	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	0000 0000	-000 0000		
71Ah	COG2FIS1 ⁽²⁾	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	--00 0000	-000 0000		
71Bh	COG2FSIM0 ⁽²⁾	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	0000 0000	-000 0000		
71Ch	COG2FSIM1 ⁽²⁾	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	--00 0000	-000 0000		
71Dh	COG2ASD0 ⁽²⁾	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—	0001 01--	0001 01--		
71Eh	COG2ASD1 ⁽²⁾	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	0000 0000	0000 0000		
71Fh	COG2STR ⁽²⁾	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	0000 0000	0000 0000		

BANK 15

78Ch — 793h	—	未実装									—	—
794h	PRG1RTSS	—	—	—	—	RTSS<3:0>				---- 0000	---- 0000	
795h	PRG1FTSS	—	—	—	—	FTSS<3:0>				---- 0000	---- 0000	
796h	PRG1INS	—	—	—	—	INS<2:0>				---- -000	---- -000	
797h	PRG1CON0	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	0-000 0000	0-00 0000	
798h	PRG1CON1	—	—	—	—	—	RDY	FPOL	RPOL	---- -000	---- -000	
799h	PRG1CON2	—	—	—	ISET<4:0>					---0 0000	---0 0000	
79Ah	PRG2RTSS ⁽²⁾	—	—	—	—	RTSS<3:0>				---- 0000	---- 0000	
79Bh	PRG2FTSS ⁽²⁾	—	—	—	—	FTSS<3:0>				---- 0000	---- 0000	
79Ch	PRG2INS ⁽²⁾	—	—	—	—	INS<2:0>				---- -000	---- -000	
79Dh	PRG2CON0 ⁽²⁾	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	0-000 0000	0-00 0000	
79Eh	PRG2CON1 ⁽²⁾	—	—	—	—	—	RDY	FPOL	RPOL	---- -000	---- -000	
79Fh	PRG2CON2 ⁽²⁾	—	—	—	ISET<4:0>					---0 0000	---0 0000	

BANK 16 ~ 26

x0Ch/ x8Ch — x1Fh/ x9Fh	—	未実装									—	—
-------------------------------------	---	-----	--	--	--	--	--	--	--	--	---	---

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR 時の値	その他のリセット時の値
BANK 27											
D8Ch D8Dh	—	未実装								—	—
D8Eh	PWMEN	—	—	MPWM6EN ⁽²⁾	MPWM5EN	—	—	—	—	--00 ----	--00 ----
D8Fh	PWMLD	—	—	MPWM6LD ⁽²⁾	MPWM5LD	—	—	—	—	--00 ----	--00 ----
D90h	PWMOUT	—	—	MPWM6OUT ⁽²⁾	MPWM5OUT	—	—	—	—	--00 ----	--00 ----
D91h	PWM5PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
D92h	PWM5PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
D93h	PWM5DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
D94h	PWM5DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
D95h	PWM5PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
D96h	PWM5PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
D97h	PWM5OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
D98h	PWM5OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
D99h	PWM5TMRL	TMR<7:0>								0000 0000	0000 0000
D9Ah	PWM5TMRH	TMR<15:8>								0000 0000	0000 0000
D9Bh	PWM5CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
D9Ch	PWM5INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 0000	---- 0000
D9Dh	PWM5INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 0000	---- 0000
D9Eh	PWM5CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 --00	-000 --00
D9Fh	PWM5LDCON	LDA	LDT ⁽²⁾	—	—	—	—	—	LDS ⁽²⁾	00-- ----0	00-- ----0
DA0h	PWM5OFCON	—	OFM<1:0> ⁽²⁾		OFO	—	—	—	OFS ⁽²⁾	-000 ----0	-000 ----0
DA1h	PWM6PHL ⁽²⁾	PH<7:0>								xxxx xxxx	uuuu uuuu
DA2h	PWM6PHH ⁽²⁾	PH<15:8>								xxxx xxxx	uuuu uuuu
DA3h	PWM6DCL ⁽²⁾	DC<7:0>								xxxx xxxx	uuuu uuuu
DA4h	PWM6DCH ⁽²⁾	DC<15:8>								xxxx xxxx	uuuu uuuu
DA5h	PWM6PRL ⁽²⁾	PR<7:0>								xxxx xxxx	uuuu uuuu
DA6h	PWM6PRH ⁽²⁾	PR<15:8>								xxxx xxxx	uuuu uuuu
DA7h	PWM6OFL ⁽²⁾	OF<7:0>								xxxx xxxx	uuuu uuuu
DA8h	PWM6OFH ⁽²⁾	OF<15:8>								xxxx xxxx	uuuu uuuu
DA9h	PWM6TMRL ⁽²⁾	TMR<7:0>								0000 0000	0000 0000
DAAh	PWM6TMRH ⁽²⁾	TMR<15:8>								0000 0000	0000 0000
DABh	PWM6CON ⁽²⁾	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
DACH	PWM6INTE ⁽²⁾	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 0000	---- 0000
DADh	PWM6INTF ⁽²⁾	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 0000	---- 0000
DAEh	PWM6CLKCON ⁽²⁾	—	PS<2:0>			—	—	CS<1:0>		-000 --00	-000 --00
DAFh	PWM6LDCON ⁽²⁾	LDA	LDT	—	—	—	—	—	LDS	00-- ----0	00-- ----0
DB0h	PWM6OFCON ⁽²⁾	—	OFM<1:0>		OFO	—	—	—	OFS	-000 ----0	-000 ----0
DB1h ~ DBFh	—	未実装								—	—

凡例: x = 未知, u = 不変, q = 条件による, - = 未実装, 「0」として読み出し, r = 予約済み
網掛け表示されている位置は未実装, 「0」として読み出し

- Note
- 1: 未実装, 「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値
BANK 28											
E0Ch — E0Eh	—	未実装								—	—
E0Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	---- --0	---- --0
E10h	INTPPS	—	—	—	INTPPS<4:0>				---	0010	---u uuuu
E11h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>				---	0010	---u uuuu
E12h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>				---	0101	---u uuuu
E13h	T1GPPS	—	—	—	T1GPPS<4:0>				---	0100	---u uuuu
E14h	CCP1PPS	—	—	—	CCP1PPS<4:0>				---	0101	---u uuuu
E15h	CCP2PPS ⁽²⁾	—	—	—	CCP2PPS<4:0>				---	0011	---u uuuu
E16h	COG1INPPS	—	—	—	COG1INPPS<4:0>				---	0010	---u uuuu
E17h	COG2INPPS ⁽²⁾	—	—	—	COG2INPPS<4:0>				---	0010	---u uuuu
E18h	—	未実装								—	—
E19h	T2CKIPPS	—	—	—	T2CKIPPS<4:0>				---	0101	---u uuuu
E1Ah	T3CKIPPS	—	—	—	T3CKIPPS<4:0>				---	0101	---u uuuu
E1Bh	T3GPPS	—	—	—	T3GPPS<4:0>				---	0100	---u uuuu
E1Ch	T4CKIPPS	—	—	—	T4CKIPPS<4:0>				---	0001	---u uuuu
E1Dh	T5CKIPPS	—	—	—	T5CKIPPS<4:0>				---	0000	---u uuuu
E1Eh	T5GPPS	—	—	—	T5GPPS<4:0>				---	0011	---u uuuu
E1Fh	T6CKIPPS	—	—	—	T6CKIPPS<4:0>				---	0011	---u uuuu
E20h	SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>				---	0000 ⁽³⁾	---u uuuu
		—	—	—	SSPCLKPPS<4:0>				---	1110 ⁽²⁾	---u uuuu
E21h	SSPDATPPS	—	—	—	SSPDATPPS<4:0>				---	0001 ⁽³⁾	---u uuuu
		—	—	—	SSPDATPPS<4:0>				---	1100 ⁽²⁾	---u uuuu
E22h	SSPSSPPS	—	—	—	SSPSSPPS<4:0>				---	0011 ⁽³⁾	---u uuuu
		—	—	—	SSPSSPPS<4:0>				---	0110 ⁽²⁾	---u uuuu
E23h	—	未実装								—	—
E24h	RXPPS	—	—	—	RXPPS<4:0>				---	0101 ⁽³⁾	---u uuuu
		—	—	—	RXPPS<4:0>				---	1101 ⁽²⁾	---u uuuu
E25h	CKPPS	—	—	—	CKPPS<4:0>				---	0100 ⁽³⁾	---u uuuu
		—	—	—	CKPPS<4:0>				---	1111 ⁽²⁾	---u uuuu
E26h	—	未実装								—	—
E27h	—	未実装								—	—
E28h	CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>				---	0011	---u uuuu
E29h	CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>				---	0100	---u uuuu
E2Ah	CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>				---	0001	---u uuuu
E2Bh	CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>				---	0101	---u uuuu
E2Ch	PRG1RPPS	—	—	—	PRG1RPPS<4:0>				---	0100	---u uuuu
E2Dh	PRG1FPPS	—	—	—	PRG1FPPS<4:0>				---	0101	---u uuuu
E2Eh	PRG2RPPS ⁽²⁾	—	—	—	PRG2RPPS<4:0>				---	0100	---u uuuu
E2Fh	PRG2FPPS ⁽²⁾	—	—	—	PRG2FPPS<4:0>				---	0101	---u uuuu
E30h	MD1CHPPS	—	—	—	MD1CHPPS<4:0>				---	0011	---u uuuu

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note 1: 未実装、「1」として読み出します。
2: PIC16(L)F1768/9のみです。
3: PIC16(L)F1764/5のみです。
4: PIC16LF1764/5/8/9では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK28(続き)												
E31h	MD1CLPPS	—	—	—	MD1CLPPS<4:0>			---	0 100	---	u uuuu	
E32h	MD1MODPPS	—	—	—	MD1MODPPS<4:0>			---	0 101	---	u uuuu	
E33h	MD2CHPPS ⁽²⁾	—	—	—	MD2CHPPS<4:0>			---	0 011	---	u uuuu	
E34h	MD2CLPPS ⁽²⁾	—	—	—	MD2CLPPS<4:0>			---	0 100	---	u uuuu	
E35h	MD2MODPPS ⁽²⁾	—	—	—	MD2MODPPS<4:0>			---	0 101	---	u uuuu	
E36h ~ E7Fh	—	未実装							—	—	—	—

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note 1: 未実装、「1」として読み出します。
2: PIC16(L)F1768/9 のみです。
3: PIC16(L)F1764/5 のみです。
4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値
BANK 29											
E8Ch E8Fh	—	未実装								—	—
E90h	RA0PPS	—	—	—	RA0PPS<4:0>			---	0 0000	---	u uuuu
E91h	RA1PPS	—	—	—	RA1PPS<4:0>			---	0 0000	---	u uuuu
E92h	RA2PPS	—	—	—	RA2PPS<4:0>			---	0 0000	---	u uuuu
E93h	—	未実装								—	—
E94h	RA4PPS	—	—	—	RA4PPS<4:0>			---	0 0000	---	u uuuu
E95h	RA5PPS	—	—	—	RA5PPS<4:0>			---	0 0000	---	u uuuu
E96h	—	未実装								—	—
E97h	—	未実装								—	—
E98h	—	未実装								—	—
E99h	—	未実装								—	—
E9Ah	—	未実装								—	—
E9Bh	—	未実装								—	—
E9Ch	RB4PPS ⁽²⁾	—	—	—	RB4PPS<4:0>			---	0 0000	---	u uuuu
E9Dh	RB5PPS ⁽²⁾	—	—	—	RB5PPS<4:0>			---	0 0000	---	u uuuu
E9Eh	RB6PPS ⁽²⁾	—	—	—	RB6PPS<4:0>			---	0 0000	---	u uuuu
E9Fh	RB7PPS ⁽²⁾	—	—	—	RB7PPS<4:0>			---	0 0000	---	u uuuu
EA0h	RC0PPS	—	—	—	RC0PPS<4:0>			---	0 0000	---	u uuuu
EA1h	RC1PPS	—	—	—	RC1PPS<4:0>			---	0 0000	---	u uuuu
EA2h	RC2PPS	—	—	—	RC2PPS<4:0>			---	0 0000	---	u uuuu
EA3h	RC3PPS	—	—	—	RC3PPS<4:0>			---	0 0000	---	u uuuu
EA4h	RC4PPS	—	—	—	RC4PPS<4:0>			---	0 0000	---	u uuuu
EA5h	RC5PPS	—	—	—	RC5PPS<4:0>			---	0 0000	---	u uuuu
EA6h	RC6PPS ⁽²⁾	—	—	—	RC6PPS<4:0>			---	0 0000	---	u uuuu
EA7h	RC7PPS ⁽²⁾	—	—	—	RC7PPS<4:0>			---	0 0000	---	u uuuu
EA8h EEFh	—	未実装								—	—

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値
BANK 30											
F0Ch — F0Eh		未実装								—	—
F0Fh	CLCDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	---- -000	---- -000
F10h	CLC1CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F11h	CLC1POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0--- xxxx	0--- uuuu
F12h	CLC1SEL0	—	—	—	D1S<4:0>				---	xxxx	---u uuuu
F13h	CLC1SEL1	—	—	—	D2S<4:0>				---	xxxx	---u uuuu
F14h	CLC1SEL2	—	—	—	D3S<4:0>				---	xxxx	---u uuuu
F15h	CLC1SEL3	—	—	—	D4S<4:0>				---	xxxx	---u uuuu
F16h	CLC1GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuu uuuu
F17h	CLC1GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuu uuuu
F18h	CLC1GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuu uuuu
F19h	CLC1GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuu uuuu
F1Ah	CLC2CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F1Bh	CLC2POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0--- xxxx	0--- uuuu
F1Ch	CLC2SEL0	—	—	—	D1S<4:0>				---	xxxx	---u uuuu
F1Dh	CLC2SEL1	—	—	—	D2S<4:0>				---	xxxx	---u uuuu
F1Eh	CLC2SEL2	—	—	—	D3S<4:0>				---	xxxx	---u uuuu
F1Fh	CLC2SEL3	—	—	—	D4S<4:0>				---	xxxx	---u uuuu
F20h	CLC2GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuu uuuu
F21h	CLC2GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuu uuuu
F22h	CLC2GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuu uuuu
F23h	CLC2GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuu uuuu
F24h	CLC3CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F25h	CLC3POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0--- xxxx	0--- uuuu
F26h	CLC3SEL0	—	—	—	D1S<4:0>				---	xxxx	---u uuuu
F27h	CLC3SEL1	—	—	—	D2S<4:0>				---	xxxx	---u uuuu
F28h	CLC3SEL2	—	—	—	D3S<4:0>				---	xxxx	---u uuuu
F29h	CLC3SEL3	—	—	—	D4S<4:0>				---	xxxx	---u uuuu
F2Ah	CLC3GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuu uuuu
F2Bh	CLC3GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuu uuuu
F2Ch	CLC3GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuu uuuu
F2Dh	CLC3GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuu uuuu
F2Eh — F6Fh		未実装								—	—

凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

PIC16(L)F1764/5/8/9

表 3-16: 特殊機能レジスタのまとめ (続き)

アドレス	レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	その他のリセット時の値	
BANK 31												
F8Ch ~ FE3h	—	未実装								—	—	
FE4h	STATUS_SHAD	—	—	—	—	—	Z	DC	C	---- -xxx	---- -uuu	
FE5h	WREG_SHAD	ワーキングレジスタ、シャドウ								xxxx xxxx	uuuu uuuu	
FE6h	BSR_SHAD	—	—	—	バンクセレクトレジスタ、シャドウ					---x xxxx	---u uuuu	
FE7h	PCLATH_SHAD	—	プログラムカウンタラッチ、上位レジスタ、シャドウ								-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	間接データメモリアドレスポインタ 0、下位バイト、シャドウ								xxxx xxxx	uuuu uuuu	
FE9h	FSR0H_SHAD	間接データメモリアドレスポインタ 0、上位バイト、シャドウ								xxxx xxxx	uuuu uuuu	
FEAh	FSR1L_SHAD	間接データメモリアドレスポインタ 1、下位バイト、シャドウ								xxxx xxxx	uuuu uuuu	
FEBh	FSR1H_SHAD	間接データメモリアドレスポインタ 1、上位バイト、シャドウ								xxxx xxxx	uuuu uuuu	
FECh	—	未実装								—	—	
FEDh	STKPTR	—	—	—	現在のスタックポインタ					---1 1111	---1 1111	
FEEh	TOSL	Top of Stack 下位バイト								xxxx xxxx	uuuu uuuu	
FEFh	TOSH	—	Top of Stack 上位バイト								-xxx xxxx	-uuu uuuu

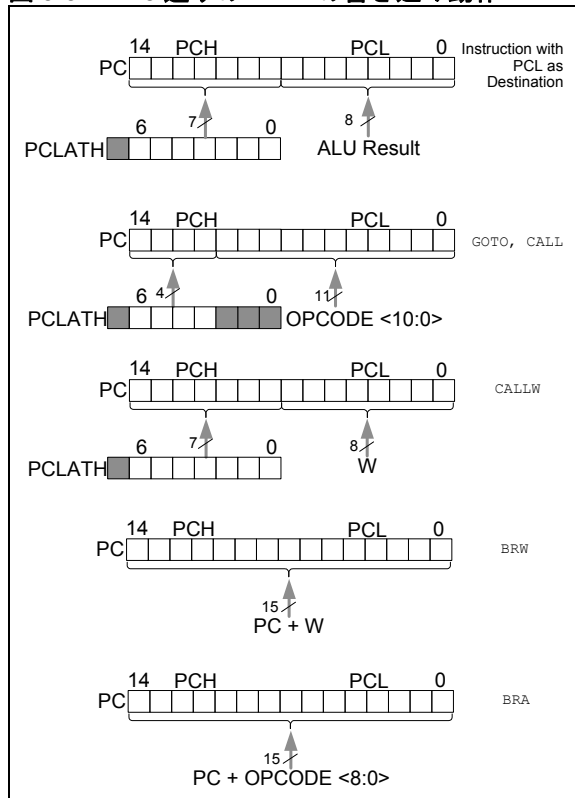
凡例: x = 未知、u = 不変、q = 条件による、- = 未実装、「0」として読み出し、r = 予約済み
網掛け表示されている位置は未実装、「0」として読み出し

- Note
- 1: 未実装、「1」として読み出します。
 - 2: PIC16(L)F1768/9 のみです。
 - 3: PIC16(L)F1764/5 のみです。
 - 4: PIC16LF1764/5/8/9 では未実装です。

3.5 PCL と PCLATH

プログラムカウンタ (PC) は 15 ビット幅です。プログラムカウンタの下位バイトには、読み書き可能なレジスタである PCL レジスタの値が格納されます。上位バイト (PC<14:8>) には PCLATH の値が格納され、これらは直接読み書きできません。PC は全種類のリセットでクリアされます。図 3-3 に、PC への書き込みにおける 5 通りの状況を示します。

図 3-3: 5 通りの PC への書き込み動作



3.5.1 PCL の書き換え

PCL レジスタを格納先とする命令を実行すると、同時にプログラムカウンタ PC<14:8> ビット (PCH) が PCLATH レジスタの内容で置き換えられます。このため、上位 7 ビットを PCLATH レジスタに書き込む事で、プログラムカウンタの内容全体を変更できます。下位 8 ビットを PCL レジスタに書き込むと、プログラムカウンタの 15 ビット全てが PCLATH レジスタに格納された値と PCL レジスタに書き込む値に変更されます。

3.5.2 計算型 GOTO

計算型 GOTO は、プログラムカウンタにオフセットを加算する事で実行されます (ADDWF PCL)。計算型 GOTO 方式を使ってテーブル読み出しを実行する場合、PCL のメモリ境界 (各 256 バイトブロック) を越えたテーブル位置へのアクセスには注意が必要です。詳細は、アプリケーションノート AN556 『Implementing a Table Read』 (DS00556) を参照してください。

3.5.3 計算型関数呼び出し

計算型の関数 CALL を使うと、プログラムで関数テーブルを維持し、ステートマシンまたはルックアップテーブルを実行するもう 1 つの方法が得られます。計算型の関数 CALL を使ってテーブル読み出しを実行する場合、PCL のメモリ境界 (各 256 バイトブロック) を越えたテーブル位置へのアクセスには注意が必要です。

CALL 命令を使う場合、PCH<2:0> と PCL レジスタには CALL 命令のオペランドが書き込まれます。PCH<6:3> には PCLATH<6:3> が書き込まれます。

CALLW 命令は、PCLATH と W を組み合わせて格納先アドレスを形成する事で計算型呼び出しを実現します。計算型 CALLW は、W レジスタにアドレスを書き込んでから CALLW を実行します。PCL レジスタには W の値が書き込まれ、PCH には PCLATH の値が書き込まれます。

3.5.4 分岐

分岐命令は PC にオフセットを加算します。これにより、再配置可能コードとページ境界をまたぐコードが可能です。分岐には 2 種類 (BRW、BRA) あります。どちらの場合も、PC がインクリメントして次の命令をフェッチします。どちらも PCL メモリの境界を越える事が可能です。

BRW を使う場合、W レジスタに符号なしアドレスを書き込んでから BRW を実行します。PC は PC + 1 + W のアドレスから読み込んだ値です。

BRA を使う場合、PC の値は (PC + 1 + BRA 命令の符号付きオペランド) です。

PIC16(L)F1764/5/8/9

3.6 スタック

全てのデバイスが 16 段 x 15 ビット幅のハードウェアスタックを搭載しています (図 3-1 参照)。スタック空間は、プログラムとデータ空間のどちらにも属しません。CALL または CALLW 命令が実行された場合、または割り込みによって分岐が発生した場合、PC の値がスタックにプッシュされます。RETURN、RETLW、RETFIE 命令のどれかが実行されると、スタックから値がポップされます。PCLATH はプッシュまたはポップの影響を受けません。

STVREN ビット (コンフィグレーションワード) を「0」にプログラムすると、スタックはリングバッファとして機能します。つまり、スタックが 16 回プッシュされると、17 回目にプッシュされた値は、1 回目のプッシュで格納された値を上書きします。18 回目のプッシュでは 2 回目のプッシュ値が上書きされます (以降同様)。リセットが有効であるかどうかに関係なく、オーバーフロー / アンダーフローが発生すると STKOVF と STKUNF フラグビットがセットされます。

Note: PUSH または POP と呼ばれる命令 / ニーモニックはありません。これらは、CALL、CALLW、RETURN、RETLW、RETFIE 命令の実行時、または割り込みアドレスへのベクタ処理時に発生する動作を指しています。

3.6.1 スタックへのアクセス

スタックへのアクセスには、TOSH、TOSL、STKPTR レジスタを使います。STKPTR は、スタックポインタの現在値を示します。TOSH:TOSL レジスタペアは、スタックの TOS (Top of Stack) を示します。どちらのレジスタも読み書き可能です。PC は 15 ビットであるため、TOS は TOSH と TOSL に分割されます。スタックへアクセスするには、STKPTR 値を調節して TOSH:TOSL を決定し、TOSH:TOSL への読み書きを実行します。STKPTR は 5 ビットでオーバーフローとアンダーフローを検出できます。

Note: 割り込み有効時の STKPTR 書き換えには注意が必要です。

通常動作中、STKPTR は CALL、CALLW、割り込み発生時にインクリメントし、RETLW、RETURN、RETFIE 発生時にデクリメントします。STKPTR をチェックする事で、いつでもスタックの空き容量を確認できます。STKPTR は、常にスタック内で使用中の場所を示します。従って、CALL または CALLW 命令は STKPTR をインクリメントしてから PC への書き込みを実行し、RETURN 命令は PC の値を読み出してから STKPTR をデクリメントします。

スタックへのアクセス例については、図 3-4 ~ 図 3-7 を参照してください。

図 3-4: スタックへのアクセス例 1

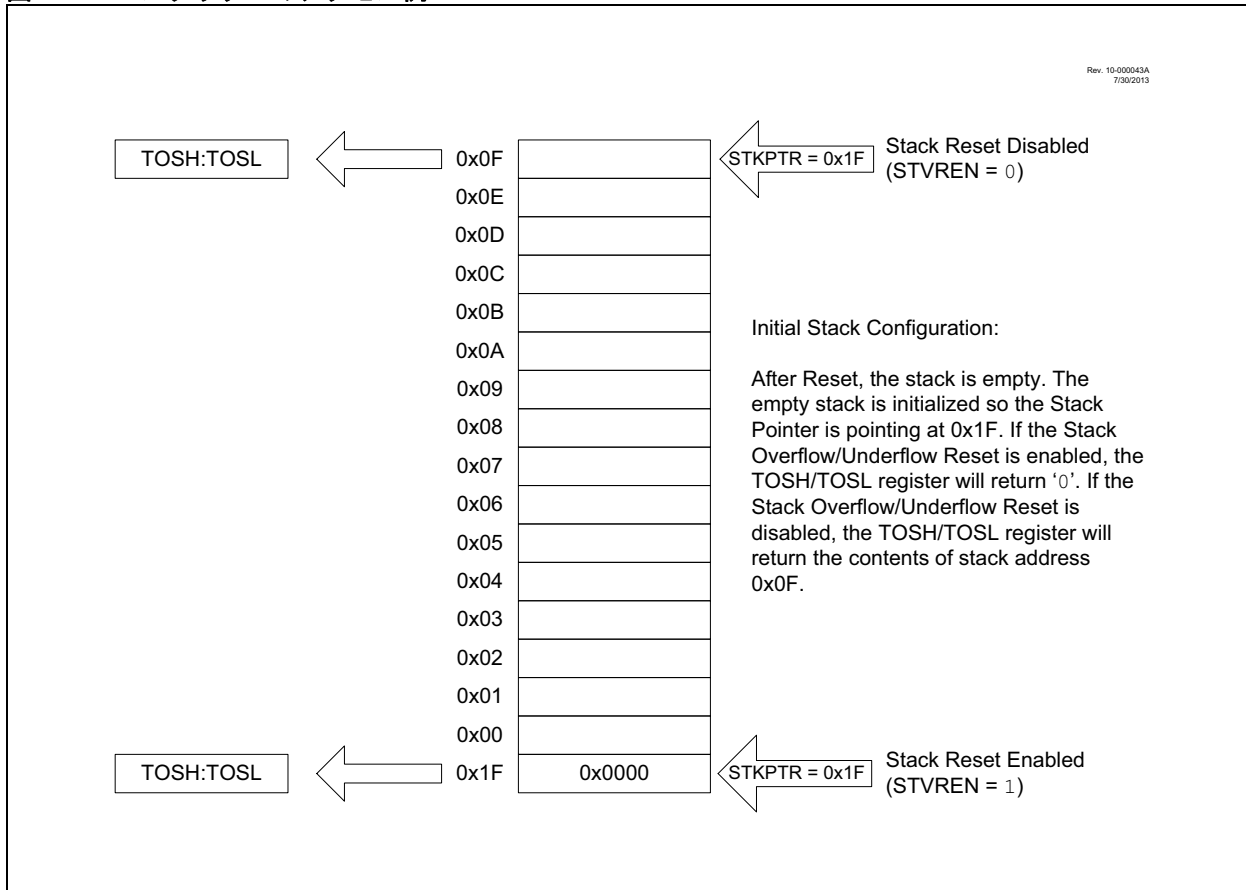


図 3-5: スタックへのアクセス例 2

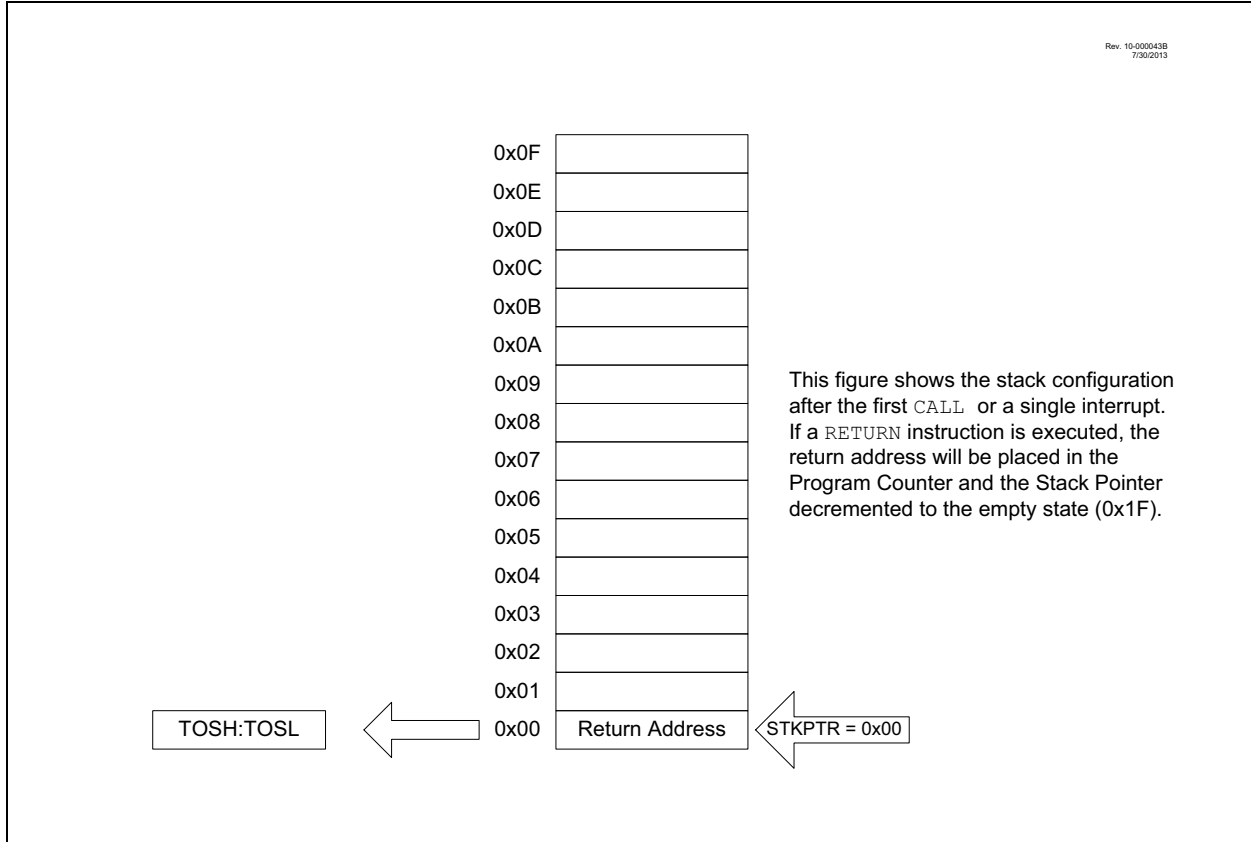
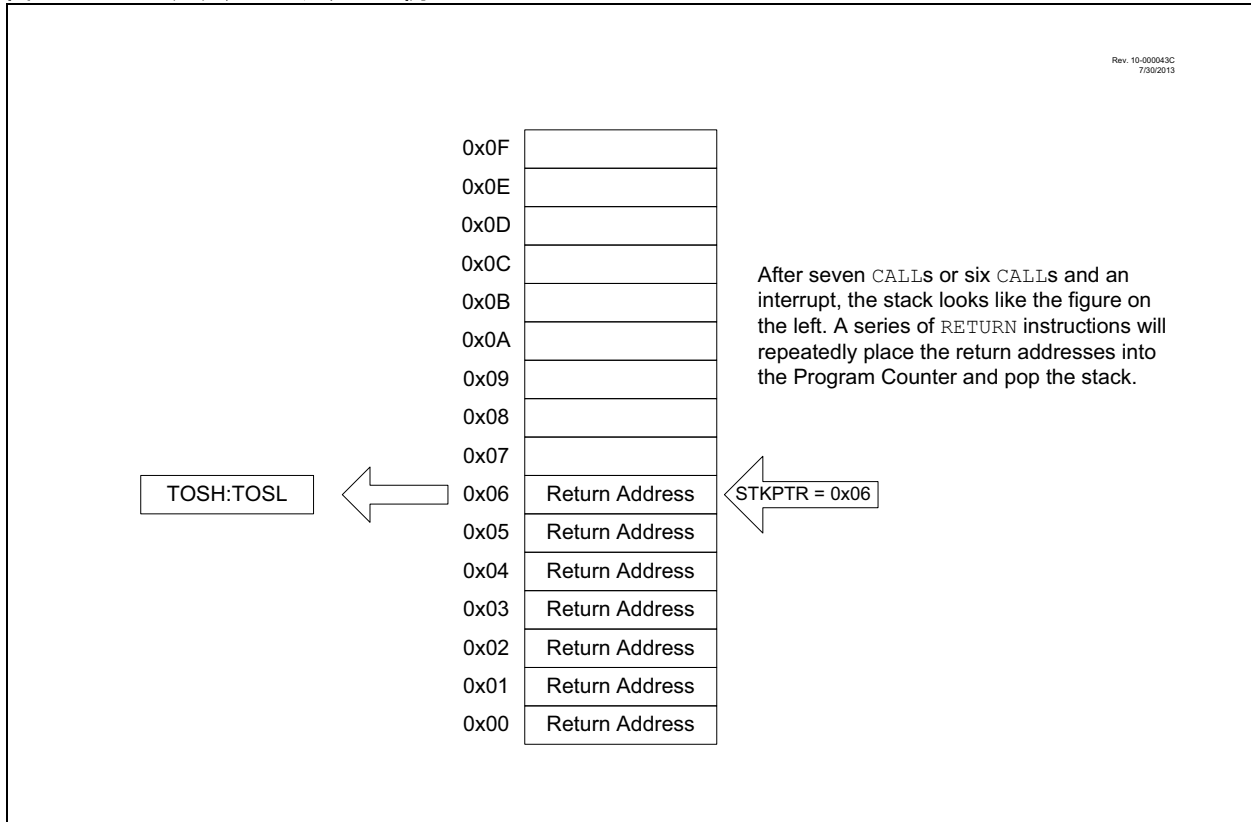
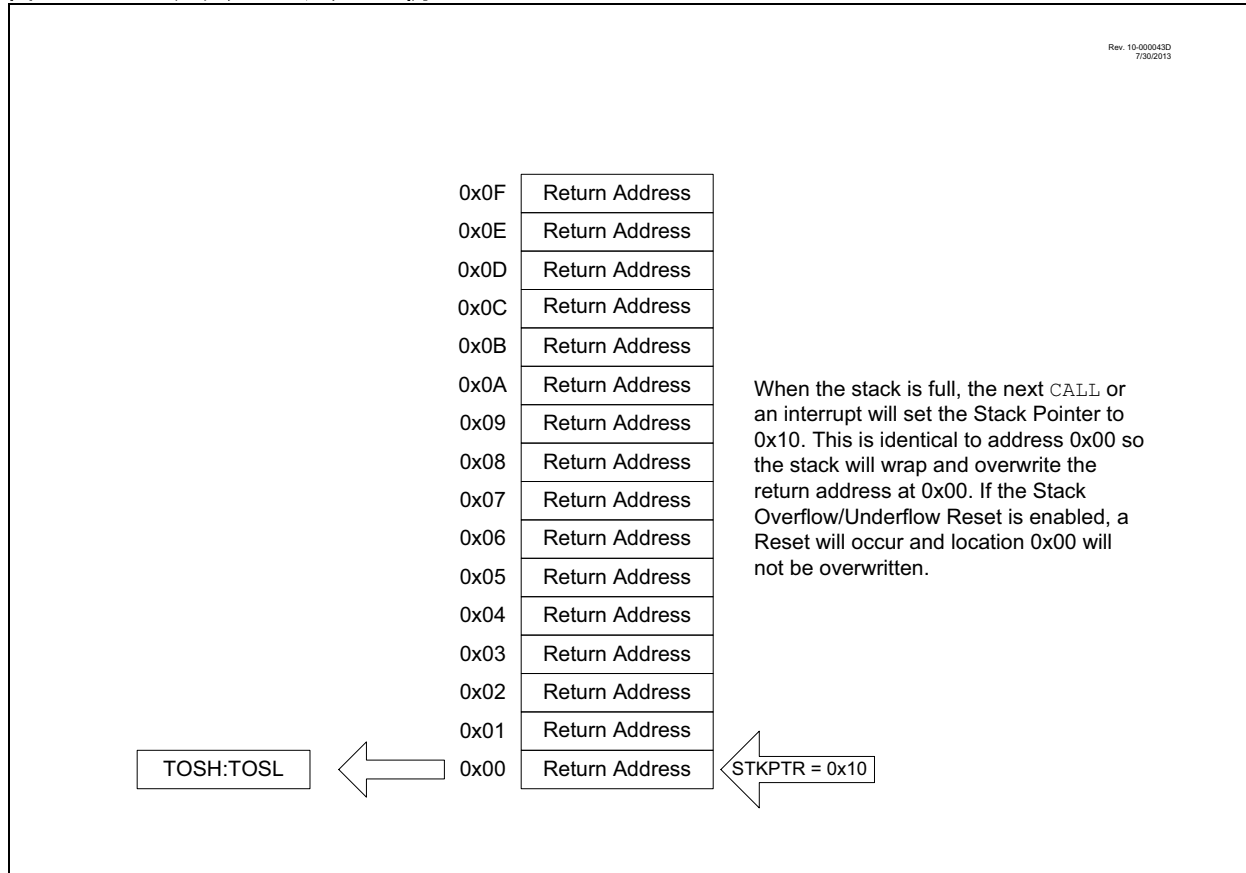


図 3-6: スタックへのアクセス例 3



PIC16(L)F1764/5/8/9

図 3-7: スタックへのアクセス例 4



3.6.2 オーバーフロー/アンダーフローによるリセット

コンフィグレーションワードのSTVRENビットが「1」にプログラムされている場合、スタックで 16 レベルを超えるプッシュ動作または 1 レベルを下回るポップ動作が生じるとデバイスがリセットされ、PCON レジスタの対応するビット（それぞれ STKOVF または STKUNF）がセットされます。

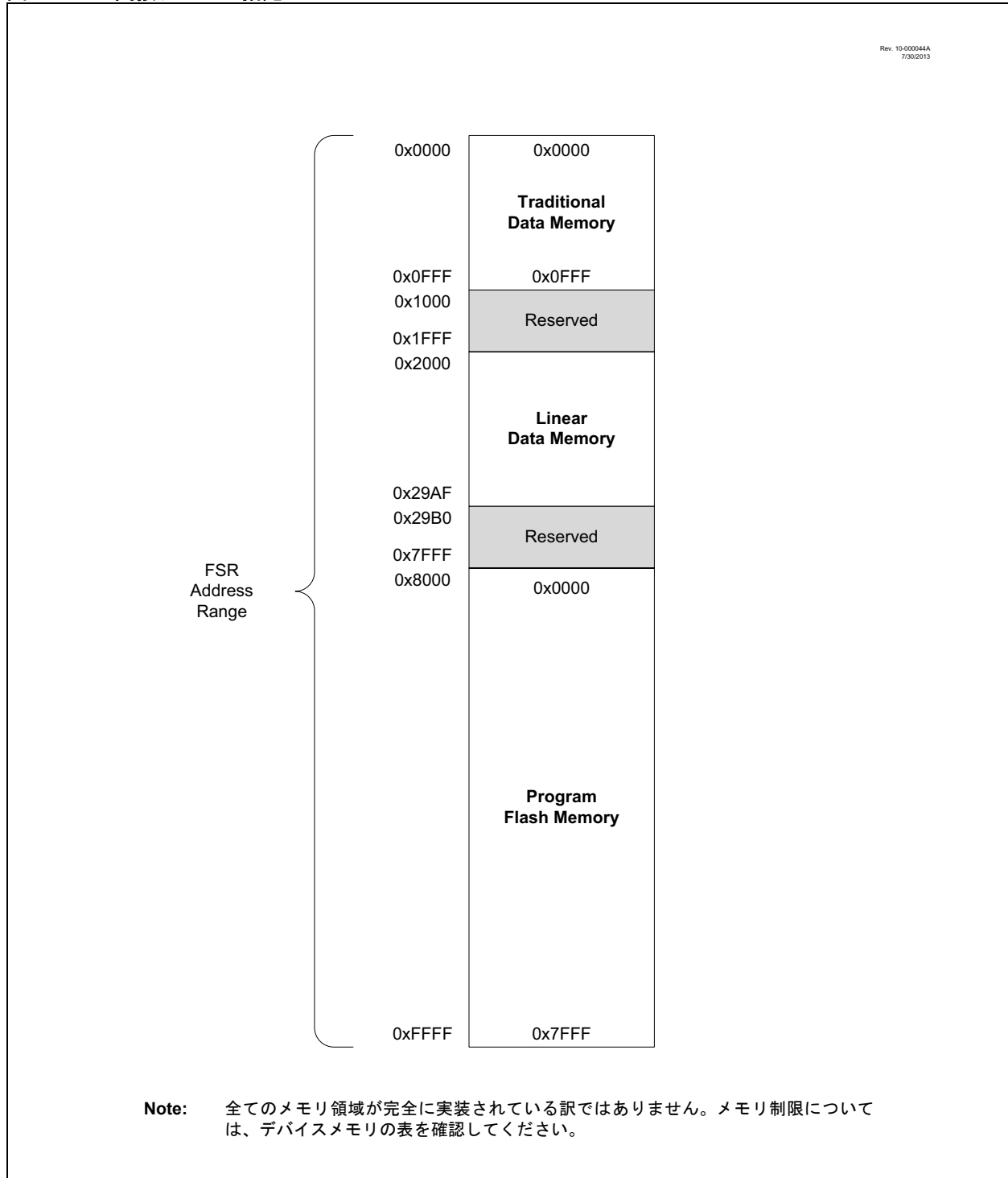
3.7 間接アドレス指定

INDFn レジスタは物理的なレジスタではありません。INDFn レジスタへアクセスする命令は、実際にはFSR（ファイルセレクト レジスタ）で指定したアドレスにあるレジスタ値へアクセスしています。FSRn アドレスが 2 つのINDFn レジスタのどちらかを示す場合、読み出し動作は「0」を返し、書き込み動作は生じません（ステータスビットは影響を受ける可能性があります）。FSRn レジスタの値は、FSRnH と FSRnL のレジスタペアで作成されます。

FSR レジスタは 16 ビットアドレスを形成し、65536ヶ所のアドレス空間があります。これらのアドレスは以下に示す 3 つのメモリ領域に分割されています。

- 従来型データメモリ
- リニア データメモリ
- プログラム フラッシュメモリ

図 3-8: 間接アドレス指定

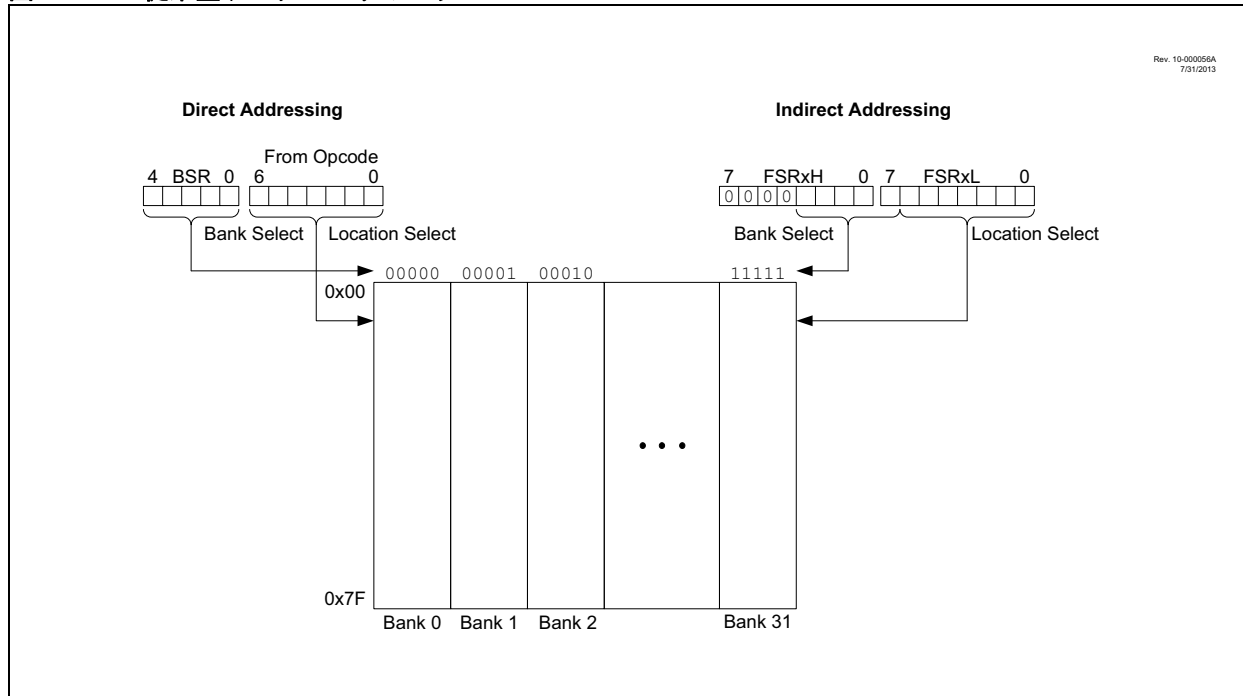


PIC16(L)F1764/5/8/9

3.7.1 従来型データメモリ

従来型データメモリの領域は、FSR アドレスの 0x000 から 0xFFFF です。このアドレスは、全ての SFR/GPR/ 共通レジスタの絶対アドレスに該当します。

図 3-9: 従来型データメモリのマップ



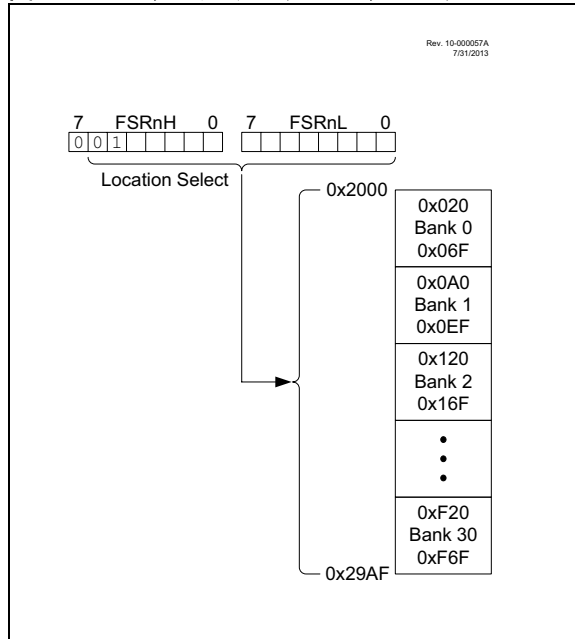
3.7.2 リニア データメモリ

リニア データメモリの領域は、FSRアドレスの0x2000から 0x29AF です。これは、全バンク内の GPR メモリ (80 バイトブロック) を示す仮想領域です。

未実装メモリの読み出しは 0x00 が出力されます。リニア データメモリ領域を使うと、バンクをまたいで FSR をインクリメントして次のバンクの GPR メモリへ直接アクセスするため、80 バイト以上のバッファとして使えます。

16 バイトの共通メモリは、リニア データメモリ領域内には含まれません。

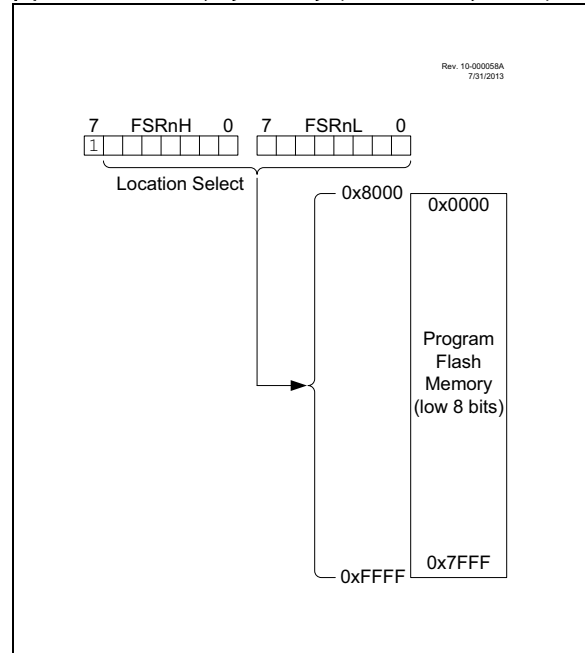
図 3-10: リニア データメモリのマップ



3.7.3 プログラム フラッシュメモリ

定数データアクセスを容易にするために、プログラムフラッシュメモリ全体がFSRアドレス空間の上位半分に割り当てられています。FSRnH の MSb がセットされている場合、下位 15 ビットがプログラムメモリのアドレスとなり、INDF でアクセスされます。ただし、INDF でアクセスできる場所は、各メモリ位置の下位 8 ビットのみです。FSR/INDF インターフェイスからはプログラムフラッシュメモリに書き込む事はできません。FSR/INDF インターフェイスを使ってプログラムフラッシュメモリへアクセスする命令を実行する場合、常に 1 命令サイクルが追加が必要です。

図 3-11: プログラムフラッシュメモリのマップ



PIC16(L)F1764/5/8/9

4.0 デバイス コンフィグレーション

デバイス コンフィグレーションは、コンフィグレーション ワード、コード保護、デバイス ID から構成されます。

4.1 コンフィグレーション ワード

デバイスには、オシレータやメモリ保護の各種設定を選択できる複数のコンフィグレーション ワードビットがあります。これらは、コンフィグレーション ワード1 (8007h)とコンフィグレーション ワード2 (8008h)として実装されています。

Note: コンフィグレーション ワードの <code>DEBUG</code> ビットは、デバッガやプログラマ等のデバイス開発ツールによって自動的に管理されます。デバイスの通常動作中は、このビットを「1」に保つ必要があります。

4.2 レジスタ定義：コンフィグレーションワード

レジスタ 4-1: CONFIG1: コンフィグレーションワード 1

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
CP ⁽¹⁾	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>		
bit 7							bit 0

凡例：

R = 読み出し可能ビット P = プログラム可能ビット U = 未実装ビット、「1」として読み出し
 「0」= ビットはクリア 「1」= ビットはセット -n = ブランクまたはバルク消去後の値

- bit 13 **FCMEN:** フェイルセーフクロックモニタ イネーブルビット
 1 = ON フェイルセーフクロックモニタと内部/外部切り換えの両方を有効にする
 0 = OFF フェイルセーフクロックモニタを無効にする
- bit 12 **IESO:** 内部/外部切り換えビット
 1 = ON 内部/外部切り換えモードを有効にする
 0 = OFF 内部/外部切り換えモードを無効にする
- bit 11 **CLKOUTEN:** クロック出カイネーブルビット
FOSC コンフィグレーションビットが LP、XT、HS モードに設定されている場合：
 このビットを無視し CLKOUT 機能を無効にする。CLKOUT ピンはオシレータとして機能する
その他の全ての FOSC モードの場合：
 1 = ON CLKOUT 機能を無効にする。CLKOUT ピンは I/O として機能する
 0 = OFF CLKOUT ピンの CLKOUT 機能を有効にする
- bit 10-9 **BOREN<1:0>:** ブラウンアウト リセット イネーブルビット
 11 = ON BOR を有効にする
 10 = NSLEEP BOR を動作時は有効、スリープ中は無効にする
 01 = SBODEN BOR を BORCON レジスタの SBODEN ビットで制御する
 00 = OFF BOR を無効にする
- bit 8 未実装：「1」として読み出し
- bit 7 **CP:** コード保護ビット⁽¹⁾
 1 = OFF プログラムメモリのコード保護を無効にする
 0 = ON プログラムメモリのコード保護を有効にする
- bit 6 **MCLRE:** MCLR/VPP ピン機能選択ビット
LVP ビット = 1 の場合：
 このビットは無視される
LVP ビット = 0 の場合：
 1 = ON MCLR/VPP ピンの機能は MCLR、弱プルアップを有効にする
 0 = OFF MCLR/VPP ピンはデジタル入力として機能する。MCLR を内部で無効にし、弱プルアップは WPUA3 ビットで制御する
- bit 5 **PWRTE:** パワーアップ タイマ イネーブルビット
 1 = OFF PWRT を無効にする
 0 = ON PWRT を有効にする
- bit 4-3 **WDTE<1:0>:** ウォッチドッグ タイマ イネーブルビット
 11 = ON WDT を有効にする
 10 = NSLEEP WDT を動作時は有効、スリープ中は無効にする
 01 = SWDTEN WDT を WDTCON レジスタの SWDTEN ビットで制御する
 00 = OFF WDT を無効にする

PIC16(L)F1764/5/8/9

レジスタ 4-1: CONFIG1: コンフィグレーションワード1 (続き)

bit 2-0	FOSC<2:0> : オシレータ選択ビット
111 = ECH	外部クロック、高消費電力モード: OSC1/CLKIN ピンに CLKIN を供給する
110 = ECM	外部クロック、中消費電力モード: OSC1/CLKIN ピンに CLKIN を供給する
101 = ECL	外部クロック、低消費電力モード: OSC1/CLKIN ピンに CLKIN を供給する
100 = INTOSC	内部 HFINTOSCCLKIN ピンは I/O として機能する
011 = EXTRC	CLKIN ピンに外部 RC 回路を接続する
010 = HS	OSC1 ピンと OSC2 ピンの間に高速水晶 / セラミック振動子を接続する
001 = XT	OSC1 ピンと OSC2 ピンの間に水晶 / セラミック振動子を接続する
000 = LP	OSC1 ピンと OSC2 ピンの間に低消費電力水晶振動子を接続する

Note 1: 消去中にコード保護を OFF にすると、フラッシュ プログラムメモリ全体が消去されます。バルク消去プログラムメモリ コマンドを実行すると、プログラム フラッシュメモリ全体とコンフィグレーションメモリが消去されます。

レジスタ 4-2: CONFIG2: コンフィグレーションワード 2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
LVP ⁽¹⁾	$\overline{\text{DEBUG}}^{\text{(2)}}$	$\overline{\text{LPBOR}}$	BORV ⁽³⁾	STVREN	PLLEN
bit 13					bit 8

R/P-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1	R/P-1
$\overline{\text{ZCD}}$	—	—	—	—	PPS1WAY	WRT<1:0>	
bit 7						bit 0	

凡例:

R = 読み出し可能ビット P = プログラム可能ビット U = 未実装ビット、「1」として読み出し
「0」= ビットはクリア 「1」= ビットはセット -n = ブランクまたはバルク消去後の値

- bit 13 **LVP:** 低電圧プログラミング イネーブルビット⁽¹⁾
1 = ON 低電圧プログラミングを有効にする
0 = OFF プログラミング時に MCLR に高電圧を印加する必要がある
- bit 12 **DEBUG:** インサーキット デバッグモード ビット⁽²⁾
1 = OFF インサーキット デバッグを無効にする。ICSPCLK と ICSPDAT は汎用 I/O ピンとして機能する
0 = ON インサーキット デバッグを有効にする。ICSPCLK と ICSPDAT はデバッグ専用である
- bit 11 **LPBOR:** 低消費電力 BOR イネーブルビット
1 = OFF 低消費電力ブラウンアウト リセットを無効にする
0 = ON 低消費電力ブラウンアウト リセットを有効にする
- bit 10 **BORV:** ブラウンアウト リセット電圧選択ビット⁽³⁾
1 = LO ブラウンアウト リセット電圧 (V_{BOR}) のトリップポイントを低に設定する
0 = HI ブラウンアウト リセット電圧 (V_{BOR}) のトリップポイントを高に設定する
- bit 9 **STVREN:** スタック オーバーフロー/アンダーフロー リセット イネーブルビット
1 = ON スタックのオーバーフローまたはアンダーフローによってリセットを発生させる
0 = OFF スタックのオーバーフローまたはアンダーフローによってリセットを発生させない
- bit 8 **PLLEN:** PLL イネーブルビット
1 = ON 4xPLL を有効にする
0 = OFF 4xPLL を無効にする
- bit 7 **ZCD:** ZCD イネーブルビット
1 = OFF ZCD を無効にする。ZCD を有効にするには、ZCDCON の ZCDSEN ビットをセットする
0 = ON ZCD を常に有効にする
- bit 6-3 **未実装:** 「1」として読み出し
- bit 2 **PPS1WAY:** PPSLOCK ビット単方向セット イネーブルビット
1 = ON PPSLOCK ビットは、ロック解除シーケンスを実行後に一度のみセットできる。一度 PPSLOCK が
セットされると、その後 PPS レジスタは変更できない
0 = OFF ロック解除シーケンスを実行すれば何度でも PPSLOCK ビットをセット/クリアできる
- bit 1-0 **WRT<1:0>:** フラッシュメモリ自己書き込み保護ビット
4 kW のフラッシュメモリ (PIC16(L)F1764/8)
11 = OFF 書き込み保護を無効にする
10 = BOOT 0000h ~ 1FFh を書き込み保護し、0200h ~ 0FFFh を PMCON 制御で変更可能にする
01 = HALF 0000h ~ 07FFh を書き込み保護し、0800h ~ 0FFFh を PMCON 制御で変更可能にする
00 = ALL 0000h ~ 0FFFh を書き込み保護し、PMCON 制御によるアドレス変更を不可にする
8 kW のフラッシュメモリ (PIC16(L)F1765/9)
11 = OFF 書き込み保護を無効にする
10 = BOOT 0000h ~ 1FFh を書き込み保護し、0200h ~ 1FFFh を PMCON 制御で変更可能にする
01 = HALF 0000h ~ 0FFFh を書き込み保護し、1000h ~ 1FFFh を PMCON 制御で変更可能にする
00 = ALL 0000h ~ 1FFFh を書き込み保護し、PMCON 制御によるアドレス変更を不可にする

- Note** 1: LVP を使ってプログラミング モードに入る場合、LVP ビットは「0」にプログラムできません。
2: コンフィグレーションワードの $\overline{\text{DEBUG}}$ ビットは、デバッガやプログラマ等のデバイス開発ツールによって自動的に管理されます。デバイスの通常動作中は、このビットを「1」に保つ必要があります。
3: 具体的なトリップポイント電圧については V_{BOR} パラメータを参照してください。

PIC16(L)F1764/5/8/9

4.3 コード保護

コード保護を使うと、不正なアクセスからデバイスを保護できます。プログラムメモリ保護は独立して制御されます。プログラムメモリへの内部アクセスは、コード保護のいかなる設定の影響も受けません。

4.3.1 プログラムメモリ保護

コンフィグレーションワードの \overline{CP} ビットによって、プログラムメモリ空間全体が外部の読み書きから保護されます。 $\overline{CP} = 0$ の場合、プログラムメモリに対する外部の読み書きが禁止され、読み出し動作には全て「0」が返されます。保護ビットの設定に関わらず、CPUは常にプログラムメモリを読み出す事ができます。プログラムメモリへの書き込みの可否は、書き込み保護設定で決まります。詳細は[セクション 4.4「書き込み保護」](#)を参照してください。

4.4 書き込み保護

書き込み保護を使うと、意図しない自己書き込みからデバイスを保護できます。ブートローダソフトウェア等のアプリケーションを保護する一方、プログラムメモリの他の部分に対する変更を許可できます。

コンフィグレーションワードの $WRT<1:0>$ ビットによって、保護するプログラムメモリブロックのサイズを定義します。

4.5 ユーザ ID

ユーザがチェックサムまたはその他のコード識別番号を格納できる ID 位置として、4 つのメモリ位置 (8000h ~ 8003h) が指定されています。これらの位置には、通常動作時に読み書きが可能です。これらのメモリ位置へのアクセスに関する詳細は、[セクション 10.4「ユーザ ID、デバイス ID、コンフィグレーションワードへのアクセス」](#)を参照してください。チェックサム計算の詳細は、『PIC16(L)F170X Memory Programming Specification』(DS40001683)を参照してください。

4.6 デバイス ID とリビジョン ID

14 ビットデバイス ID ワードは 8006h にあり、14 ビットリビジョン ID は 8005h にあります。これらの位置は読み出し専用で、消去または書き込みはできません。これらのメモリ位置へのアクセスに関する詳細は、[セクション 10.4「ユーザ ID、デバイス ID、コンフィグレーションワードへのアクセス」](#) を参照してください。

デバイス ID とリビジョン ID の読み出しには、デバイス プログラマやデバッガ等の開発ツールを使います。

4.7 レジスタ定義：デバイスとリビジョン

レジスタ 4-3: DEVID: デバイス ID レジスタ

R	R	R	R	R	R
DEV<13:8>					
bit 13			bit 8		

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット

「1」= ビットはセット 「0」= ビットはクリア

bit 13-0 DEVID<13:0>: デバイス ID ビット

デバイス	DEVID<13:0> の値
PIC16F1764	11 0000 1000 0000 (3080h)
PIC16F1765	11 0000 1000 0001 (3081h)
PIC16F1768	11 0000 1000 0100 (3084h)
PIC16F1769	11 0000 1000 0101 (3085h)
PIC16LF1764	11 0000 1000 0010 (3082h)
PIC16LF1765	11 0000 1000 0011 (3083h)
PIC16LF1768	11 0000 1000 0110 (3086h)
PIC16LF1769	11 0000 1000 0111 (3087h)

PIC16(L)F1764/5/8/9

レジスタ 4-4: REVID: リビジョン ID レジスタ

R	R	R	R	R	R	R	R
REV<13:8>							
bit 13				bit 8			

R	R	R	R	R	R	R	R
REV<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット

「1」= ビットはセット 「0」= ビットはクリア

bit 13-0 **REV<13:0>**: リビジョン ID ビット

5.0 オシレータ モジュール (フェイルセーフ クロック モニタ機能付き)

5.1 概要

オシレータ モジュールは各種のクロック源と優れた特長を備えているため、消費電力を最小限に抑え最大限の性能を実現しつつ幅広いアプリケーションに適用できます。図 5-1 に、オシレータ モジュールのブロック図を示します。

クロック源は、外部オシレータ、水晶振動子、セラミック振動子、RC(抵抗/コンデンサ)回路によって供給できます。また、2種類の内部オシレータの1つとPLL回路からシステムクロック源を供給でき、動作速度はソフトウェアで選択できます。その他のクロックの特長として、以下が挙げられます。

- ソフトウェアによるシステムクロック源選択機能 (外部か内部かを選択)
- 外部オシレータの起動からコード実行開始までのレイテンシを最小限に抑える2段階起動モード
- 外部クロック源(LP、XT、HS、ECH、ECM、ECL、EXTRCのいずれかのモード)の障害を検出して自動的に内部オシレータに切り換えるフェイルセーフクロックモニタ(FSCM)機能
- 水晶振動子によるクロック源の安定性を確保するオシレータ起動タイマ(OST)

オシレータ モジュールは、以下のクロックモードのいずれかに設定できます。

1. ECL – 外部クロック低消費電力モード (0 ~ 0.5 MHz)
2. ECM – 外部クロック中消費電力モード (0.5 ~ 4 MHz)
3. ECH – 外部クロック高消費電力モード (4 ~ 32 MHz)
4. LP – 32 kHz 低消費電力水晶振動子モード
5. XT – 中ゲイン水晶/セラミック振動子オシレータモード (最大 4 MHz)
6. HS – 高ゲイン水晶/セラミック振動子モード (4 ~ 20 MHz)
7. EXTRC – 外付け抵抗/コンデンサ
8. INTOSC – 内部オシレータ (31 kHz ~ 32 MHz)

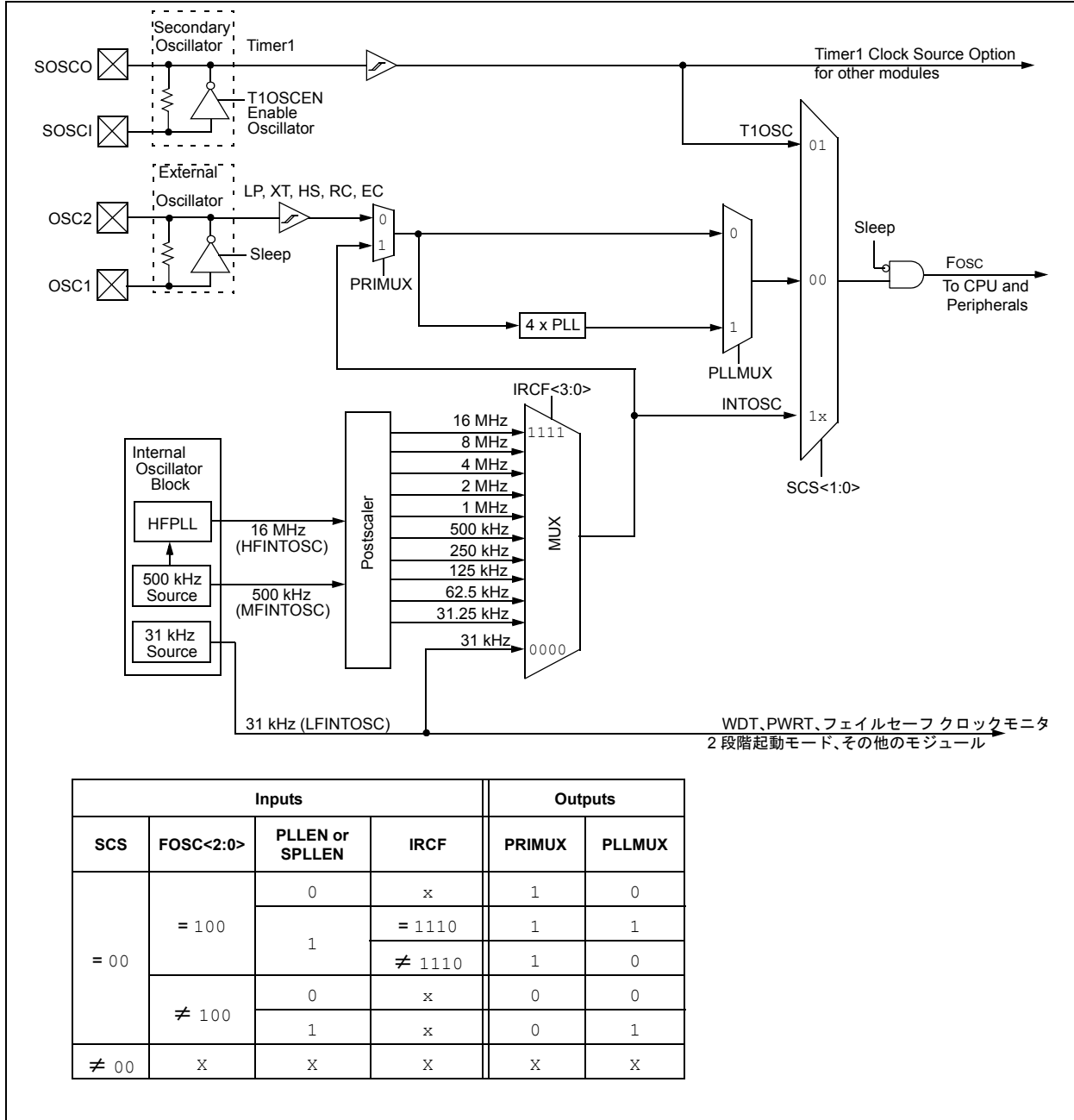
クロック源のモードは、コンフィグレーションワードの FOSC<2:0> ビットで選択します。電源投入時に使われるオシレータのタイプは FOSC ビットで決まります。

ECH、ECM、ECL クロックモードでは、デバイスのクロック源として外部の論理レベル信号を使います。LP、XT、HS クロックモードでは、外付けの水晶振動子等を使う必要があります。各モードは、それぞれ異なる周波数レンジに対して最適化されています。EXTRC クロックモードの場合、オシレータの周波数を設定する外付けの抵抗とコンデンサが必要です。

INTOSC 内部オシレータ ブロックは、LFINTOSC、MFINTOSC、HFINTOSC と呼ばれる低周波、中周波、高周波のクロック源を生成します (内部オシレータ ブロック、図 5-1 参照)。これら3つのクロック源から幅広いデバイスクロック周波数を供給できます。

PIC16(L)F1764/5/8/9

図 5-1: PIC® MCU クロック源の概略ブロック図



5.2 クロック源の種類

クロック源は外部と内部に分類されます。

外部クロック源は外部回路によって動作します。例えば、オシレータ モジュール (ECH、ECM、ECL モード)、水晶またはセラミック振動子 (LP、XT、HS モード)、EXTRC (抵抗 / コンデンサ) モード回路等です。

内部クロック源はオシレータ モジュール内部に統合されています。内部オシレータ ブロックには 2 つの内部オシレータと、1 つの専用位相ロックループ (HFPLL) があり、これらによって 3 つの内部システムクロック源、すなわち 16 MHz の高周波数内部オシレータ (HFINTOSC)、500 kHz の中周波数内部オシレータ (MFINTOSC)、31 kHz の低周波数内部オシレータ (LFINTOSC) を生成します。

システムクロックに外部クロック源と内部クロック源のどちらを使うかは、OSCCON レジスタのシステムクロック選択 (SCS) ビットで選択します。詳細は [セクション 5.3「クロック切り換え」](#) を参照してください。

5.2.1 外部クロック源

以下の操作のどちらかにより、外部クロック源をデバイスのシステムクロックとして使えます。

- コンフィグレーションワードの FOSC<2:0> ビットを外部クロック源を選択するようにプログラムします。これによってデバイスリセット時の既定値のシステムクロックとして外部クロック源が使われます。
- OSCCON レジスタの SCS<1:0> ビットに書き込む事で、システムクロック源を以下のどちらかに切り換えます。
 - セカンダリ オシレータ (実行時)
 - FOSC ビットの値によって決まる外部クロック源

詳細は [セクション 5.3「クロック切り換え」](#) を参照してください。

5.2.1.1 EC モード

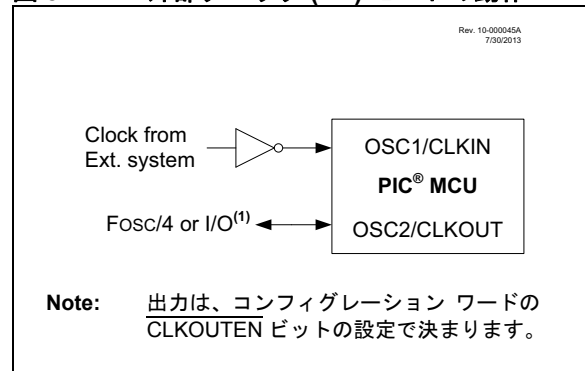
外部クロック (EC) モードでは、外部で生成された論理レベル信号をシステムクロック源として使えます。このモードで動作させる場合、外部クロック源は OSC1 入力に接続します。OSC2/CLKOUT は、汎用 I/O または CLKOUT として使えます。図 5-2 に、EC モードのピン接続を示します。

EC モードには以下の 3 つの消費電力モードがあり、コンフィグレーションワードによって選択できます。

- ECH – 高消費電力、4 ~ 32 MHz
- ECM – 中消費電力、0.5 ~ 4 MHz
- ECL – 低消費電力、0 ~ 0.5 MHz

EC モードを選択した場合、オシレータ起動タイム (OST) は無効です。従って、パワーオンリセット (POR) 後またはスリープからの復帰後に動作の遅延は発生しません。PIC® MCU は完全なスタティック設計であるため、外部クロック入力を停止すると、全てのデータがそのままの状態ではデバイスが動作を中断します。外部クロックが再開すると、デバイスは停止直後の状態から動作を再開します。

図 5-2: 外部クロック (EC) モードの動作



5.2.1.2 LP、XT、HS モード

LP、XT、HS モードでは、水晶振動子またはセラミック振動子を OSC1 と OSC2 に接続して使います (図 5-3)。これら 3 つのモードによって、内部インバータアンプのゲインが低 / 中 / 高に設定され、種類と速度の異なる振動子に対応します。

LP オシレータモードでは、内部インバータアンプのゲインが最も低く設定されます。3 つのモードのうち消費電流が最も少ないのが LP モードです。このモードは、32.768 kHz の音叉型水晶振動子 (腕時計用水晶振動子) の駆動専用です。

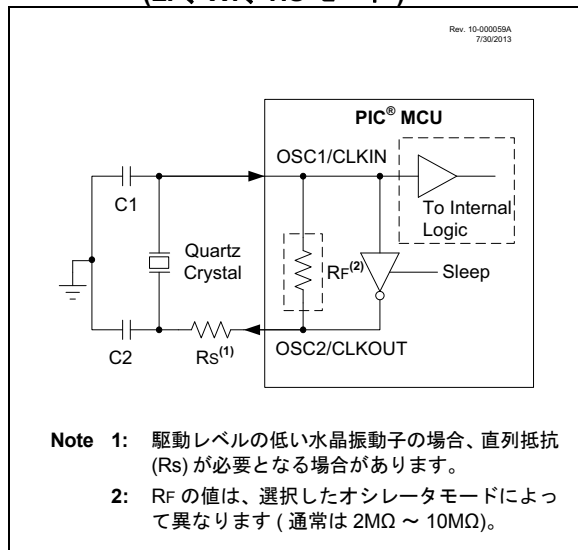
XT オシレータモードでは、内部インバータアンプのゲインが中程度に設定されます。消費電流は 3 つのモードの中間です。このモードは、駆動レベル仕様が中程度の振動子に最も適しています。

HS オシレータモードでは、内部インバータアンプのゲインが最も高く設定されます。消費電流は 3 つのモードのうち最大です。このモードは、駆動レベルを高く設定する必要のある振動子に最適です。

図 5-3 と 図 5-4 に、それぞれ水晶振動子とセラミック振動子を使った代表的な回路を示します。

PIC16(L)F1764/5/8/9

図 5-3: 水晶振動子による動作
(LP、XT、HS モード)

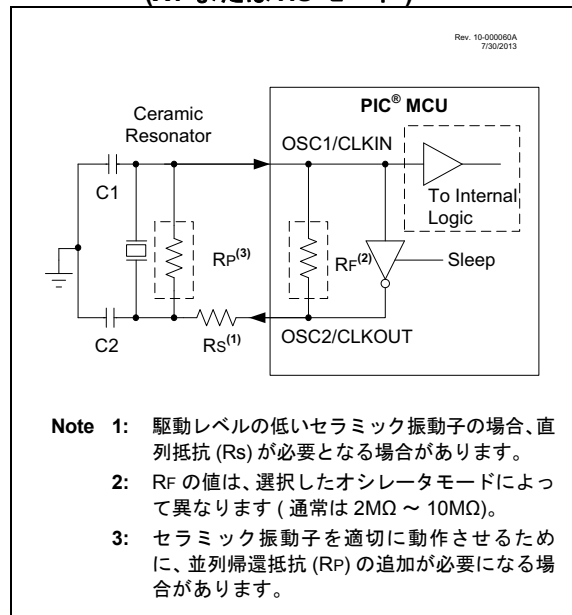


Note 1: 水晶振動子の特性は、種類、パッケージ、メーカーにより異なります。仕様と推奨アプリケーションについては、各メーカーのデータシートを参照してください。

- アプリケーションで予測される V_{DD} と仕様温度でオシレータの性能を必ず検証します。
- オシレータ設計時の参考として、Microchip 社の以下のアプリケーションノートを参照してください。

- AN826 『Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices』 (DS00826)
- AN849 『Basic PIC® Oscillator Design』 (DS00849)
- AN943 『Practical PIC® Oscillator Analysis and Design』 (DS00943)
- AN949 『Making Your Oscillator Work』 (DS00949)

図 5-4: セラミック振動子の動作
(XT または HS モード)



5.2.1.3 オシレータ起動タイマ (OST)

オシレータ モジュールが LP、XT、HS モードのどれかに設定されている場合、オシレータ起動タイマ (OST) は OSC1 の発振を 1024 回カウントします。これは、パワーオン リセット (POR) 後のパワーアップ タイマ (PWRT) のタイムアウト時 (設定されている場合)、またはスリープからの復帰後に実行されます。この間、プログラム カウンタはインクリメントせず、プログラムの実行は保留されます。ただし、FSCM または 2 段階起動のどちらかが有効な場合を除きます。これらが有効な場合、OST のカウント中もコード実行は継続します。その場合の周波数は、選択されている INTOSC 周波数です。OST を使う事で、水晶またはセラミック振動子を使ったオシレータ回路が完全に起動し、オシレータ モジュールに安定したシステムクロックが確実に供給されます。

外部オシレータの起動からコード実行までのレイテンシを最小限に抑えるには、2 段階クロック起動モードを使います ([セクション 5.4「2 段階クロック起動モード」](#) 参照)。

5.2.1.4 4x PLL

本オシレータ モジュールは、システムクロック源を供給するための、内外部のクロック源と共に使う事ができる 4x PLL を内蔵しています。4x PLL の入力周波数は仕様レンジ内である事が必要です。表 36-9 の PLL クロック タイミング仕様を参照してください。

4x PLL は以下の2つの方法のどちらかによって有効化します。

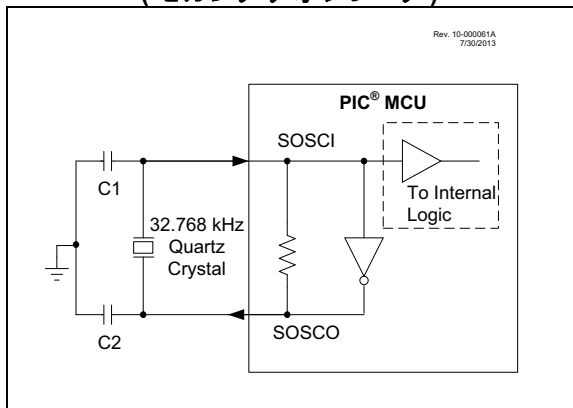
1. コンフィグレーション ワードの PLEN ビットに「1」を書き込む。
2. OSCCON レジスタの SPLLEN ビットに「1」を書き込む。(コンフィグレーション ワードの PLEN ビットに「1」が書き込まれている場合、SPLLEN の値は無視されます。)

5.2.1.5 セカンダリ オシレータ

セカンダリ オシレータは、Timer1 モジュール専用の水晶振動子用オシレータです。デバイスピンの SOSCO と SOSCI の間に接続した 32.768 kHz の水晶振動子によるタイマ動作向けに最適化されています。

セカンダリ オシレータは代替システムクロック源として使い、クロック切り換え機能によって実行中に選択できます。詳細は[セクション 5.3「クロック切り換え」](#)を参照してください。

図 5-5: 水晶振動子の動作
(セカンダリ オシレータ)



Note 1: 水晶振動子の特性は、種類、パッケージ、メーカーにより異なります。仕様と推奨アプリケーションについては、各メーカーのデータシートを参照してください。

2: アプリケーションで予測される VDD と仕様温度でオシレータの性能を必ず検証してください。

3: オシレータ設計時の参考として、Microchip 社の以下のアプリケーションノートを参照してください。

- AN826 『Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices』 (DS00826)
- AN849 『Basic PIC® Oscillator Design』 (DS00849)
- AN943 『Practical PIC® Oscillator Analysis and Design』 (DS00943)
- AN949 『Making Your Oscillator Work』 (DS00949)
- TB097 『Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS』 (DS91097)
- AN1288 『Design Practices for Low-Power External Oscillators』 (DS01288)

PIC16(L)F1764/5/8/9

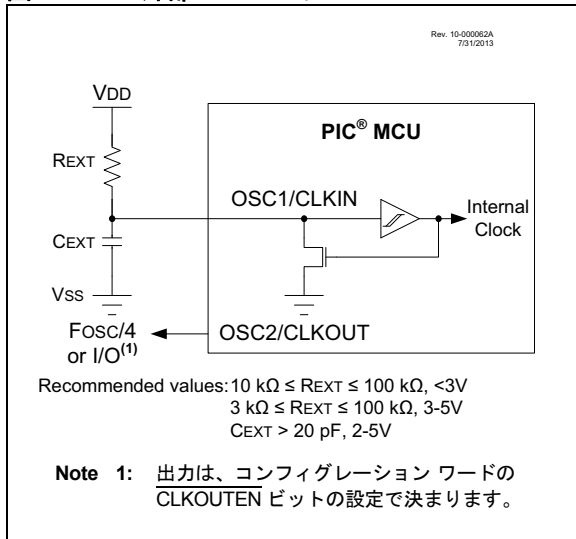
5.2.1.6 外部 RC モード

外部抵抗 / コンデンサ (EXTRC) モードでは、外付けの RC 回路を使います。クロックにそれほど高い精度が要求されない場合にこのモードを使えば、コストを最小限に抑えながら周波数選択の柔軟性を最大限に高める事ができます。

RC 回路は OSC1 へ接続します。OSC2/CLKOUT は、汎用 I/O または CLKOUT として使えます。OSC2/CLKOUT ピンの機能は、コンフィグレーションワードの CLKOUTEN ビットの設定で決まります。

図 5-6 に、外部 RC モードの接続を示します。

図 5-6: 外部 RC モード



RC オシレータの周波数は、電源電圧、抵抗 (REXT)、コンデンサ (CEXT) の値と動作温度で決まります。これ以外にオシレータの周波数に影響する要因として、以下のものがあります。

- しきい値電圧のばらつき
- 部品の製造公差
- パッケージによる静電容量のばらつき

使う外付け RC 部品の製造公差によるばらつきも考慮する必要があります。

5.2.2 内部クロック源

以下のいずれかの操作を実行する事で、内部オシレータブロックをシステムクロックとして使うように設定できます。

- コンフィグレーションワードの FOSC<2:0> ビットを INTOSC クロック源を選択するようにプログラムします。これによってデバイスリセット時に既定値のシステムクロックとしてこのクロック源が使われます。
- OSCCON レジスタの SCS<1:0> ビットに書き込む事で、実行中にシステムクロック源を内部オシレータに切り換えます。詳細は [セクション 5.3「クロック切り換え」](#) を参照してください。

INTOSC モードでは、OSC1/CLKIN は汎用 I/O として使えます。OSC2/CLKOUT は、汎用 I/O または CLKOUT として使えます。

OSC2/CLKOUT ピンの機能は、コンフィグレーションワードの CLKOUTEN ビットの設定で決まります。

内部オシレータブロックには、3つの内部システムクロック源のいずれか1つを生成するための、2つの独立したオシレータと専用位相ロックループ (HFPLL) があります。

1. **HFINTOSC**(高周波数内部オシレータ)は工場で校正されており、16 MHz で動作します。HFINTOSC クロック信号は、500 kHz の MFINTOSC ソースと専用位相ロックループ (HFPLL) から生成されます。HFINTOSC の周波数は、OSCTUNE レジスタ ([レジスタ 5-3](#)) によって、ユーザがソフトウェアで調整できます。
2. **MFINTOSC**(中周波数内部オシレータ)は工場で校正されており、500 kHz で動作します。MFINTOSC の周波数は、OSCTUNE レジスタ ([レジスタ 5-3](#)) によって、ユーザがソフトウェアで調整できます。
3. **LFINTOSC**(低周波数内部オシレータ)は未校正で、31 kHz で動作します。

5.2.2.1 HFINTOSC

高周波数内部オシレータ (HFINTOSC) は、工場で校正済みの 16 MHz の内部クロック源です。HFINTOSC の周波数は、OSCTUNE レジスタ (レジスタ 5-3) を使って、ソフトウェアで変更できます。

HFINTOSC の出力は、ポストスケーラとマルチプレクサに接続されています (図 5-1 参照)。OSCCON レジスタの IRCF<3:0> ビットを使って、HFINTOSC が生成する複数の周波数のうちいずれか 1 つをソフトウェアで選択できます。詳細は[セクション 5.2.2.7「内部オシレータのクロック切り換えタイミング」](#)を参照してください。

HFINTOSC は以下の方法で有効化します。

- OSCCON レジスタの IRCF<3:0> ビットを必要な HF 周波数に設定し、かつ
- FOSC<2:0> = 100 または
- OSCCON レジスタのシステムクロック源 (SCS) ビットを「1x」に設定する

高速起動オシレータを使う事で、HFINTOSC に切り換わる前に内部回路の電源を立ち上げ、安定化できます。

OSCSTAT レジスタの HFIOFR (高周波数内部オシレータレディ) ビットは、HFINTOSC が動作中で使用可能な状態であるかどうかを示します。

OSCSTAT レジスタの HFIOFL (高周波数内部オシレータロック状態) ビットは、HFINTOSC が最終値の 2% 以内で動作しているかどうかを示します。

OSCSTAT レジスタの HFIOFS (高周波数内部オシレータ安定) ビットは、HFINTOSC が定格値の 0.5% 以内で動作しているかどうかを示します。

5.2.2.2 MFINTOSC

中周波数内部オシレータ (MFINTOSC) は、工場で校正済みの 500 kHz の内部クロック源です。MFINTOSC の周波数は、OSCTUNE レジスタ (レジスタ 5-3) によって、ソフトウェアで変更できます。

MFINTOSC の出力は、ポストスケーラとマルチプレクサに接続されています (図 5-1 参照)。周波数は、OSCCON レジスタの IRCF<3:0> ビットを使って、MFINTOSC が生成する 9 つのうちいずれか 1 つをソフトウェアで選択できます。詳細は[セクション 5.2.2.7「内部オシレータのクロック切り換えタイミング」](#)を参照してください。

MFINTOSC は以下の方法で有効化します。

- OSCCON レジスタの IRCF<3:0> ビットを必要な HF 周波数に設定し、かつ
- FOSC<2:0> = 100 または
- OSCCON レジスタのシステムクロック源 (SCS) ビットを「1x」に設定する

OSCSTAT レジスタの MFIOFR (中周波数内部オシレータレディ) ビットは、MFINTOSC が動作中であるかどうかを示します。

5.2.2.3 内部オシレータの周波数調整

500 kHz の内部オシレータは工場で校正済みです。この内部オシレータは、OSCTUNE レジスタ (レジスタ 5-3) への書き込みによって、ソフトウェアで調整できます。HFINTOSC と MFINTOSC クロック信号は 500 kHz の内部オシレータから発生させるため、OSCTUNE レジスタの値の変更はこれら両方のクロック源に影響します。

OSCTUNE レジスタの既定値は「0」です。値は 6 ビットの 2 の補数です。値を 1Fh に設定すると最大周波数に調整されます。値を 20h に設定すると最小周波数に調整されます。

OSCTUNE レジスタを変更すると、オシレータの周波数は設定した値への移行を開始します。この移行中もコード実行は続きます。周波数が変化した事を知らせる手段は備えていません。

OSCTUNE レジスタの設定は、LFINTOSC の周波数には影響を与えません。LFINTOSC クロック源の周波数で決まる、パワーアップ タイマ (PWRT)、ウォッチドッグ タイマ (WDT)、フェイルセーフ クロック モニタ (FSCM)、周辺モジュール等の機能は、周波数の変化に影響を受けません。

5.2.2.4 LFINTOSC

低周波数内部オシレータ (LFINTOSC) は、未校正の 31 kHz 内部クロック源です。

LFINTOSC の出力は、マルチプレクサに接続されています (図 5-1 参照)。OSCCON レジスタの IRCF<3:0> ビットを使って、ソフトウェアで 31 kHz を選択します。詳細は[セクション 5.2.2.7「内部オシレータのクロック切り換えタイミング」](#)を参照してください。LFINTOSC はパワーアップ タイマ (PWRT)、ウォッチドッグ タイマ (WDT)、フェイルセーフ クロック モニタ (FSCM) の周波数としても使われます。

LFINTOSC が有効となるのは、システムクロック源として 31 kHz (OSCCON レジスタの IRCF<3:0>=000) の内部オシレータ ブロックを選択 (OSCCON レジスタの SCS ビット = 1x) するか、以下のいずれかを有効にした場合です。

- OSCCON レジスタの IRCF<3:0> ビットを必要な LF 周波数に設定し、かつ
- FOSC<2:0> = 100 または
- OSCCON レジスタのシステムクロック源 (SCS) ビットを「1x」に設定する

LFINTOSC を使う周辺モジュールは以下の通りです。

- パワーアップ タイマ (PWRT)
- ウォッチドッグ タイマ (WDT)
- フェイルセーフ クロック モニタ (FSCM)

OSCSTAT レジスタの LFIOFR (低周波数内部オシレータレディ) ビットは、LFINTOSC が動作中で使用可能な状態であるかどうかを示します。

PIC16(L)F1764/5/8/9

5.2.2.5 内部オシレータの周波数選択

システムクロック速度は、OSCCON レジスタの内部オシレータ周波数選択ビット IRCF<3:0> を使って、ソフトウェアで選択できます。

16 MHz の HFINTOSC、500 kHz の MFINTOSC、31 kHz の LFINTOSC のポストスケール出力は、マルチプレクサに接続されています (図 5-1 参照)。OSCCON レジスタの内部オシレータ周波数選択ビット IRCF<3:0> によって、内部オシレータの周波数出力を選択します。周波数は、ソフトウェアで以下から 1 つを選択できます。

- 32 MHz (4x PLL が必要)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (リセット後の既定値)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note: 全てのリセット発生後、OSCCON レジスタの IRCF<3:0> ビットは「0111」に設定され、周波数として 500 kHz が選択されます。ユーザは IRCF ビットを変更する事で、別の周波数を選択できます。

OSCCON レジスタの IRCF<3:0> ビットでは、同じ周波数で複数の選択肢があります。これらの選択肢の選び方で、システム設計のトレードオフを調整できます。同じ周波数でもオシレータ源を変える事で消費電力を低減できる場合があります。同じオシレータ源で周波数を変更すれば、遷移を高速化できます。

5.2.2.6 32 MHz 内部オシレータの周波数選択

外部オシレータ ブロック用の 4x PLL と内部オシレータ ブロックを使うと、32 MHz の内部システムクロックを生成できます。32 MHz の内部クロック源を使うには、以下の設定が必要です。

- デバイスのシステムクロックとして INTOSC ソースを使うため、コンフィグレーションワードの FOSC ビットを設定する (FOSC<2:0> = 100)。
- コンフィグレーションワードの FOSC<2:0> によって決まるクロックを使うため、OSCCON レジスタの SCS ビットをクリアする (SCS<1:0> = 00)。
- 8 MHz HFINTOSC を使うように、OSCCON レジスタの IRCF ビットを設定する (IRCF<3:0> = 1110)。
- 4x PLL を有効にするため、OSCCON レジスタの SPLLEN ビット、またはコンフィグレーションワードの PLLEN ビットに「1」を書き込む。

Note: コンフィグレーションワードの PLLEN ビットを使う場合、4x PLL をソフトウェアによって無効にする事はできません。また、SPLLEN は選択できません。

OSCCON レジスタの SCS ビットを「1x」に設定した場合、4x PLL と内部オシレータの組み合わせは使えません。内部オシレータと 4x PLL を組み合わせて使うには、SCS ビットを「00」に設定する必要があります。

5.2.2.7 内部オシレータのクロック切り換え タイミング

HFINTOSC、MFINTOSC、LFINTOSC 間で切り換える際、切り換え後のオシレータが省電力目的でシャットダウンされている場合があります (図 5-7 参照)。この場合、OSCCON レジスタの IRCF<3:0> ビットを変更してから周波数が切り換わるまでに遅延が生じます。HFINTOSC、MFINTOSC、LFINTOSC の各オシレータのどれが現在アクティブであるかは、OSCSTAT レジスタの値で確認できます。周波数切り換えシーケンスは以下の通りです。

1. OSCCON レジスタの IRCF<3:0> ビットが変更される。
2. 切り換え後のクロックがシャットダウンされている場合、クロック起動遅延が開始する。
3. クロック切り換え回路が、切り換え前のクロックの立ち下がりエッジを待つ。
4. 現在のクロックが Low に保持され、クロック切り換え回路が切り換え後クロックの立ち上がりエッジを待つ。
5. 切り換え後クロックがアクティブとなる。
6. 必要に応じて OSCSTAT レジスタが変更される。
7. クロックの切り換えが完了する。

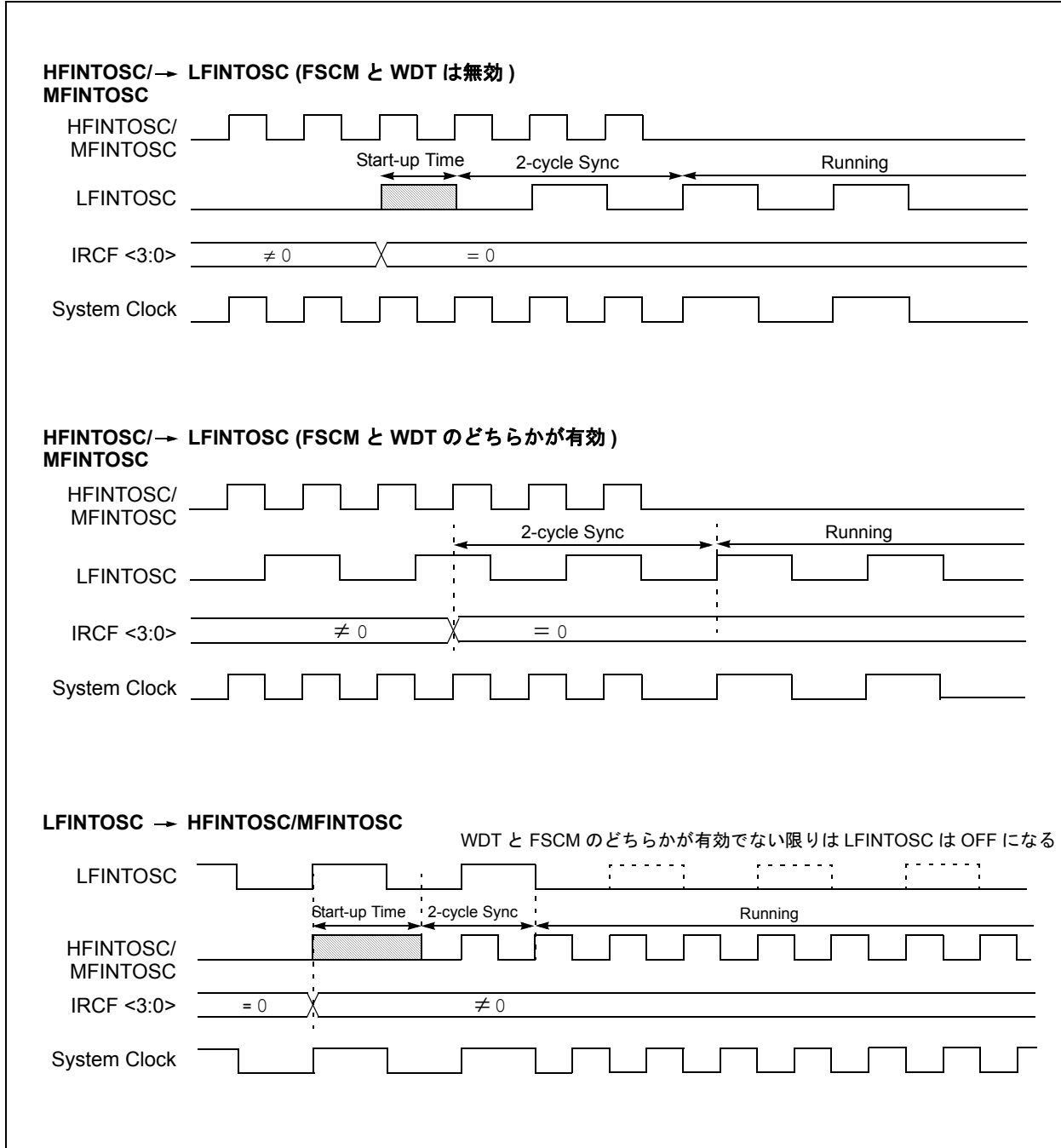
詳細は図 5-7 を参照してください。

同じクロック源で別の内部オシレータ周波数に切り換える場合、起動遅延なしで切り換わります。表 5-1 に、クロック切り換え時に発生する遅延を示します。

起動遅延の仕様は、**セクション 36.0「電氣的仕様」**のオシレータの表に記載されています。

PIC16(L)F1764/5/8/9

図 5-7: 内部オシレータの切り換えタイミング



5.3 クロック切り換え

システムクロック源として外部クロック源と内部クロック源のどちらを使うかは、OSCCON レジスタのシステムクロック選択 (SCS) ビットで設定します。SCS ビットでは以下のクロック源を選択できます。

- コンフィグレーションワードの FOSC ビットで決まる既定値のシステムオシレータ
- Timer1 32 kHz 水晶振動子
- 内部オシレータ ブロック (INTOSC)

5.3.1 システムクロック選択 (SCS) ビット

OSCCON レジスタのシステムクロック選択 (SCS) ビットは、CPU と周辺モジュールに使うシステムクロック源を選択します。

- OSCCON レジスタの SCS ビット = 00 の場合、システムクロック源はコンフィグレーションワードの FOSC<2:0> ビットの値によって決まります。
- OSCCON レジスタの SCS ビット = 01 の場合、システムクロック源はセカンダリオシレータです。
- OSCCON レジスタの SCS ビット = 1x の場合、システムクロック源には、OSCCON レジスタの IRCF<3:0> ビットで選択した内部オシレータ周波数が使われます。リセット後、OSCCON レジスタの SCS ビットは常にクリアされます。

Note: 2 段階起動またはフェイルセーフ クロックモニタにおいて自動的にクロックが切り換わった場合、OSCCON レジスタの SCS ビットは変更されません。現在のシステムクロック源は、OSCSTAT レジスタの OSTS ビットで確認できます。

クロック源を切り換えた場合、新しいクロックが安定するまで遅延が必要です。表 5-1 に、これらのオシレータの遅延を示します。

5.3.2 オシレータ起動タイマステータス (OSTS) ビット

OSCSTAT レジスタのオシレータ起動タイマステータス (OSTS) ビットは、システムクロックがコンフィグレーションワードの FOSC<2:0> ビットで定義された外部クロック源、または内部クロック源のどちらで動作中であることを示します。LP/XT/HS モードでは、OSTS ビットを参照する事でオシレータ起動タイマ (OST) がタイムアウトしたかどうかを確認できます。OSTS はセカンダリオシレータのステータスを反映していません。

5.3.3 セカンダリオシレータ

セカンダリオシレータは、Timer1 モジュール専用の水晶振動子用オシレータです。デバイスピンの SOSCO と SOSCI の間に接続した 32.768 kHz の水晶振動子によるタイマ動作向けに最適化されています。

セカンダリオシレータは、T1CON レジスタの OSCEN 制御ビットで有効にします。Timer1 周辺モジュールに関する詳細は、[セクション 22.0 「Timer1/3/5 モジュール \(ゲート制御対応\)」](#) を参照してください。

5.3.4 セカンダリオシレータ レディ (SOSCR) ビット

セカンダリオシレータをシステムクロック源として選択する前に、動作の準備が整っている事を確認する必要があります。OSCSTAT レジスタのセカンダリオシレータレディ (SOSCR) ビットは、セカンダリオシレータが使用可能な状態であるかどうかを示します。SOSCR ビットがセットされた後に、SCS ビットをセットしてセカンダリオシレータを選択できます。

PIC16(L)F1764/5/8/9

5.4 2段階クロック起動モード

2段階起動モードは、外部オシレータの起動からコード実行までのレイテンシを最小限に抑える事によって、省電力性をさらに高める働きをします。2段階起動モードは復帰に要する時間から外部オシレータの起動時間を除外できるため、特にスリープを多用するアプリケーションで総消費電力を低減できます。このモードにより、アプリケーションはスリープから復帰し、外部オシレータの安定を待たずに、クロック源として内部オシレータ ブロック INTOSC を使って少数の命令を実行し、すぐにスリープに戻る事ができますからです。

2段階起動は、オシレータ モジュールを LP、XT、HS モードのいずれかに設定した場合に有利です。これらのモードではオシレータ起動タイマ (OST) が有効であり、オシレータをシステムクロック源として使えるまで 1024 カウント待機する必要があります。

オシレータ モジュールが LP、XT、HS 以外のモードに設定されている場合、2段階起動は無効です。外部クロック オシレータは POR 後またはスリープからの復帰時に安定化の時間を必要としないためです。

デバイスがスリープに移行する前に OST のカウント数が 1024 に達すると、OSCSTAT レジスタの OSTS ビットがセットされ、プログラム実行のクロック源が外部オシレータに切り換わります。しかし、復帰している時間が短い場合、システムが外部オシレータによって動作する機会が全くない場合もあります。

Note: SLEEP 命令を実行するとオシレータ起動時間のカウントは終了し、OSCSTAT レジスタの OSTS ビットはクリアされた状態を維持します。

5.4.1 2段階起動モードの設定

2段階起動モードは、以下のように設定します。

- コンフィグレーションワードの IESO(内部/外部切り換え) ビット = 1 にセットして、2段階起動モードを有効にする。
 - OSCCON レジスタの SCS ビット = 00 に設定する。
 - コンフィグレーションワードの FOSC<2:0> ビットを LP/XT/HS のいずれかのモードに設定する。
- 2段階起動モードには、以下の場合に移行します。
- パワーオンリセット (POR) 後、パワーアップタイマ (PWRT) が有効な場合 PWRT のタイムアウト後
 - スリープから復帰後

表 5-1: オシレータ切り換え遅延

切り換え前	切り換え後	周波数	オシレータ起動遅延
スリープ/POR	LFINTOSC ⁽¹⁾ MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31 kHz 31.25 ~ 500 kHz 31.25 kHz ~ 16 MHz	オシレータのウォームアップ遅延 (TWARM)
スリープ/POR	EC, RC ⁽¹⁾	DC ~ 32 MHz	2 サイクル
LFINTOSC	EC, RC ⁽¹⁾	DC ~ 32 MHz	1 サイクル
スリープ/POR	セカンダリ オシレータ LP, XT, HS ⁽¹⁾	32 kHz ~ 20 MHz	1024 クロックサイクル (OST)
任意のクロック源	MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31.25 ~ 500 kHz 31.25 kHz ~ 16 MHz	2 μs(概算値)
任意のクロック源	LFINTOSC ⁽¹⁾	31 kHz	1 サイクル
任意のクロック源	セカンダリ オシレータ	32 kHz	1024 クロックサイクル (OST)
PLL 無効	PLL 有効	16 ~ 32 MHz	2 ms(概算値)

Note 1: PLL 無効です。

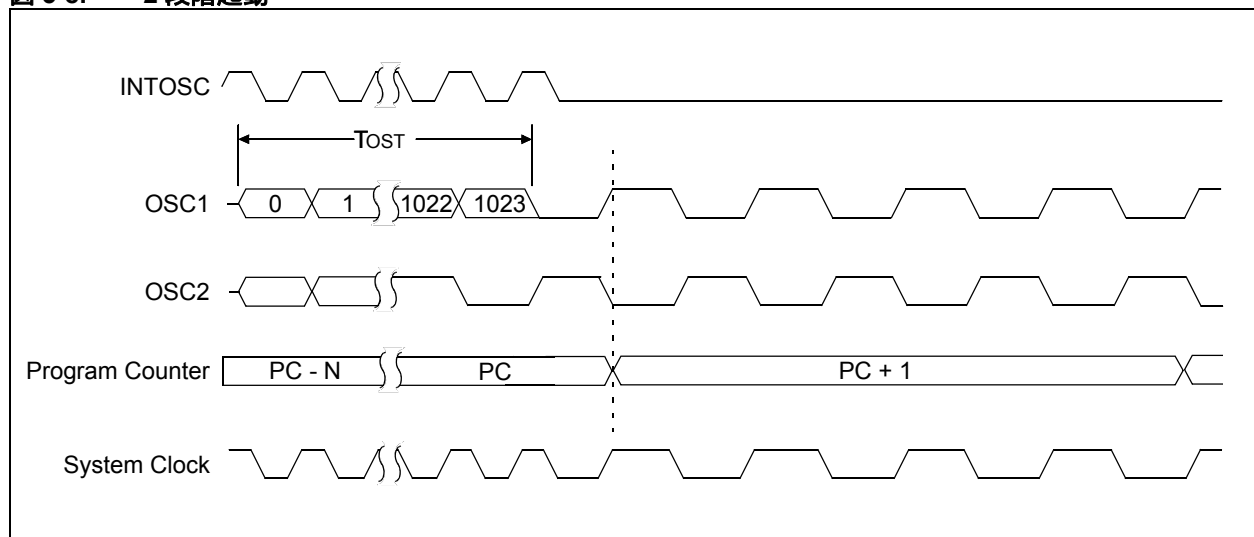
5.4.2 2段階起動のシーケンス

1. パワーオンリセットまたはスリープから復帰する。
2. OSCCONレジスタのIRCF<3:0>ビットで設定した周波数の内部オシレータで、命令の実行が始まる。
3. OST が有効になり、1024 クロックサイクルをカウントする。
4. OST がタイムアウトし、内部オシレータの立ち下がりがエッジを待つ。
5. OSTS がセットされる。
6. 切り換え後クロック (LP、XT、HS のいずれかのモード)の立ち下がりがエッジまで、システムクロックが Low に保持される。
7. システムクロックが外部クロック源に切り換わる。

5.4.3 2段階クロックステータスの確認

OSCSTAT レジスタの OSTS ビットの状態を参照すると、マイクロコントローラがコンフィギュレーションワードの FOSC<2:0> ビットで定義した外部クロック源で動作しているか、内部オシレータで動作しているかを確認できます。

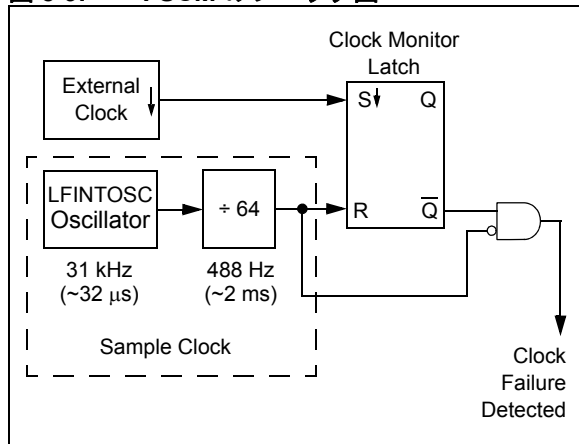
図 5-8: 2段階起動



5.5 フェイルセーフ クロックモニタ

フェイルセーフ クロックモニタ (FSCM) により、外部オシレータに障害が発生してもデバイスは動作を続ける事ができます。オシレータ起動タイマ (OST) のタイムアウト後、FSCM はいつでもオシレータの障害を検出できます。FSCM を有効にするには、コンフィグレーションワードの FCMEN ビットをセットします。FSCM は全ての外部オシレータモード (LP、XT、HS、EC、セカンダリ オシレータ、RC) に適用できます。

図 5-9: FSCM のブロック図



5.5.1 フェイルセーフ検出

FSCM モジュールは、外部オシレータと FSCM サンプルクロックを比較する事で、オシレータの障害を検出します。サンプルクロックは LFINTOSC を 64 分周して生成します。図 5-9 を参照してください。障害検出ブロック内部はラッチ回路です。外部クロックの立ち上がりエッジごとに、ラッチがセットされます。そして、サンプルクロックの立ち上がりエッジごとに、ラッチがクリアされます。サンプルクロックが 1/2 サイクル経過しても外部クロックが Low にならない場合、障害として検出されます。

5.5.2 フェイルセーフの動作

外部クロックに障害が発生すると、FSCM によってデバイスクロックが内部クロック源に切り換えられ、PIR2 レジスタの OSFIF フラグビットがセットされます。PIE2 レジスタの OSFIE ビットもセットされている場合に OSFIF フラグがセットされると、割り込みが発生します。この時点から、デバイスのファームウェアがクロック障害に対する対応を開始できます。ファームウェアが外部オシレータの再起動に成功し、再び外部クロック動作に切り換えるまでの間、システムは内部クロック源で動作を継続します。

FSCM が使う内部クロック源は、OSCCON レジスタの IRCF<3:0> ビットで決まります。従って、障害の発生前に内部オシレータを設定しておく事ができます。

5.5.3 フェイルセーフ条件の解消

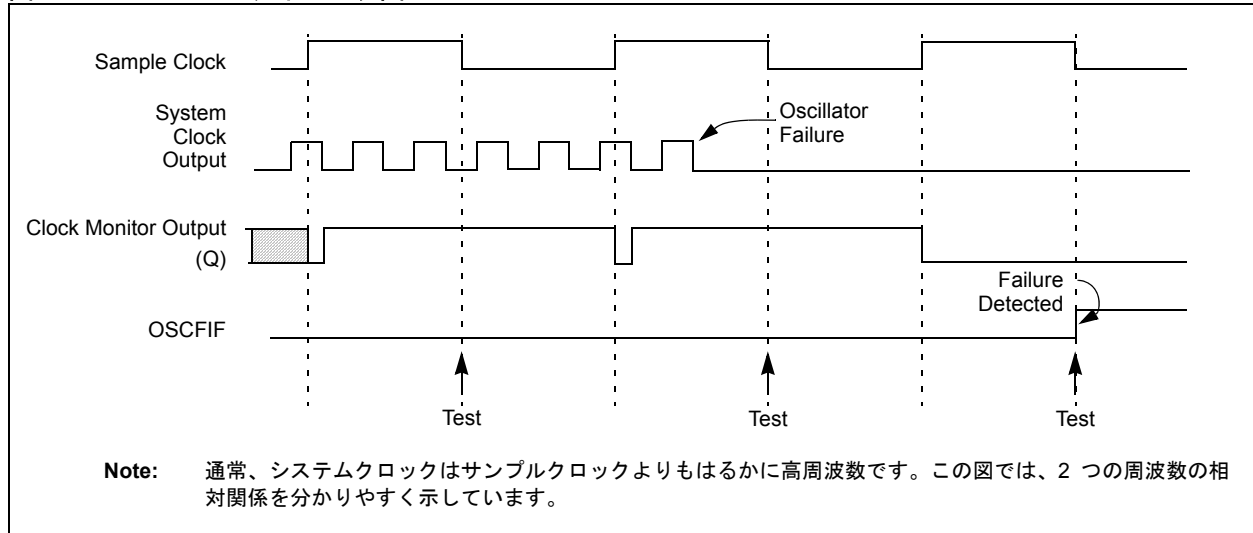
リセット、SLEEP 命令の実行、OSCCON レジスタの SCS ビットの変更によって、フェイルセーフ条件を解消できます。SCS ビットが変更されると OST が再開します。OST のカウント中、デバイスは OSCCON レジスタで選択した INTOSC で動作を継続します。OST がタイムアウトすると、外部クロック源に正常に切り換わった後にフェイルセーフ条件が解消されます。外部クロック源に切り換える前に、OSFIF ビットをクリアしておく必要があります。フェイルセーフ条件が解消されていないければ、ハードウェアによって OSFIF フラグが再度セットされます。

5.5.4 リセットまたはスリープからの復帰

FSCM は、オシレータ起動タイマ (OST) のタイムアウト後にオシレータの障害を検出するように設計されています。OST が動作するのは、スリープからの復帰後または全タイプのリセット後です。EC または RC クロックモードの場合、OST は使いません。リセット後または復帰完了後すぐに FSCM を有効にするためです。FSCM が有効な場合、2 段階起動も有効です。従って、デバイスは OST 動作中は常にコードを実行します。

Note: オシレータによって起動時間が大きく異なるため、(リセットまたはスリープからの復帰後の)オシレータ起動中は、フェイルセーフ回路はアクティブになりません。一定時間経過後、OSCSTAT レジスタのステータスビットを参照してオシレータ起動状態を確認し、システムクロックの切り換えが正常に実行された事を確認してください。

図 5-10: FSCM のタイミング図



PIC16(L)F1764/5/8/9

5.6 レジスタ定義 : オシレータ制御

レジスタ 5-1: OSCCON: オシレータ制御レジスタ

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>			—	SCS<1:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7 **SPLLEN:** ソフトウェア PLL イネーブルビット
 コンフィグレーションワードの PLEN = 1 の場合:
 SPLLEN ビットを無視する。4x PLL を常に有効にする (オシレータ要件の影響を受ける)
 コンフィグレーションワードの PLEN = 0 の場合:
 1 = 4x PLL を有効にする
 0 = 4x PLL を無効にする

bit 6-3 **IRCF<3:0>:** 内部オシレータ周波数選択ビット
 1111 = 16 MHz HF
 1110 = 8 MHz または 32 MHz HF⁽²⁾
 1101 = 4 MHz HF
 1100 = 2 MHz HF
 1011 = 1 MHz HF
 1010 = 500 kHz HF⁽¹⁾
 1001 = 250 kHz HF⁽¹⁾
 1000 = 125 kHz HF⁽¹⁾
 0111 = 500 kHz MF (リセット時の既定値)
 0110 = 250 kHz MF
 0101 = 125 kHz MF
 0100 = 62.5 kHz MF
 0011 = 31.25 kHz HF⁽¹⁾
 0010 = 31.25 kHz MF
 000x = 31 kHz LF

bit 2 **未実装:** 「0」として読み出し

bit 1-0 **SCS<1:0>:** システムクロック選択ビット
 1x = 内部オシレータ ブロック
 01 = セカンダリ オシレータ
 00 = コンフィグレーションワードの FOSC<2:0> ビットによって決まるクロック

Note 1: HFINTOSC から生成される周波数と重複しています。
Note 2: SPLLEN ビットのセット時は 32 MHz です。 [セクション 5.2.2.6 「32 MHz 内部オシレータの周波数選択」](#) を参照してください。

レジスタ 5-2: OSCSTAT: オシレータ ステータス レジスタ

R-1/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
SOSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件による

- bit 7 **SOSCR:** セカンダリ オシレータレディ ビット
T1OSCEN = 1 の場合:
 1 = セカンダリ オシレータの準備が完了している
 0 = セカンダリ オシレータの準備が完了していない
T1OSCEN = 0 の場合:
 1 = セカンダリ クロック源が常にレディ状態である
- bit 6 **PLLR:** 4x PLL レディビット
 1 = 4x PLL の準備が完了している
 0 = 4x PLL の準備が完了していない
- bit 5 **OSTS:** オシレータ起動タイムステータス ビット
 1 = コンフィグレーションワードの FOSC<2:0> ビットで定義されたクロックによって動作中である
 0 = 内部オシレータによって動作中である (FOSC<2:0> = 100)
- bit 4 **HFIOFR:** 高周波数内部オシレータレディ ビット
 1 = HFINTOSC の準備が完了している
 0 = HFINTOSC の準備が完了していない
- bit 3 **HFIOFL:** 高周波数内部オシレータロック状態ビット
 1 = HFINTOSC の精度が 2% 以内である
 0 = HFINTOSC の精度が 2% 以内でない
- bit 2 **MFIOFR:** 中周波数内部オシレータレディ ビット
 1 = MFINTOSC の準備が完了している
 0 = MFINTOSC の準備が完了していない
- bit 1 **LFIOFR:** 低周波数内部オシレータレディ ビット
 1 = LFINTOSC の準備が完了している
 0 = LFINTOSC の準備が完了していない
- bit 0 **HFIOFS:** 高周波数内部オシレータ安定状態ビット
 1 = HFINTOSC の精度が 0.5% 以内である
 0 = HFINTOSC の精度が 0.5% 以内でない

PIC16(L)F1764/5/8/9

レジスタ 5-3: OSCTUNE: オシレータ調整レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	TUN<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **TUN<5:0>:** 周波数調整ビット
100000 = 最小周波数
.
.
.
111111 =
000000 = オシレータ モジュールは工場では校正済みの周波数で動作する
000001 =
.
.
.
011110 =
011111 = 最大周波数

表 5-2: クロック源関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ内容記載ページ
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		86
OSCSTAT	SOSCR	PLLRL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	87
OSCTUNE	—	—	TUN<5:0>						88
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
T1CON	CS<1:0>		CKPS<1:0>		OSCN	SYNC	—	ON	228

凡例: — = 未実装、「0」として読み出し。網掛けの部分はクロック源では使いません。

Note 1: PIC16(L)F1768/9のみです。

表 5-3: クロック源関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ内容記載ページ
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>	FOSC<2:0>				

凡例: — = 未実装、「0」として読み出し。網掛けの部分はクロック源では使いません。

6.0 リセット

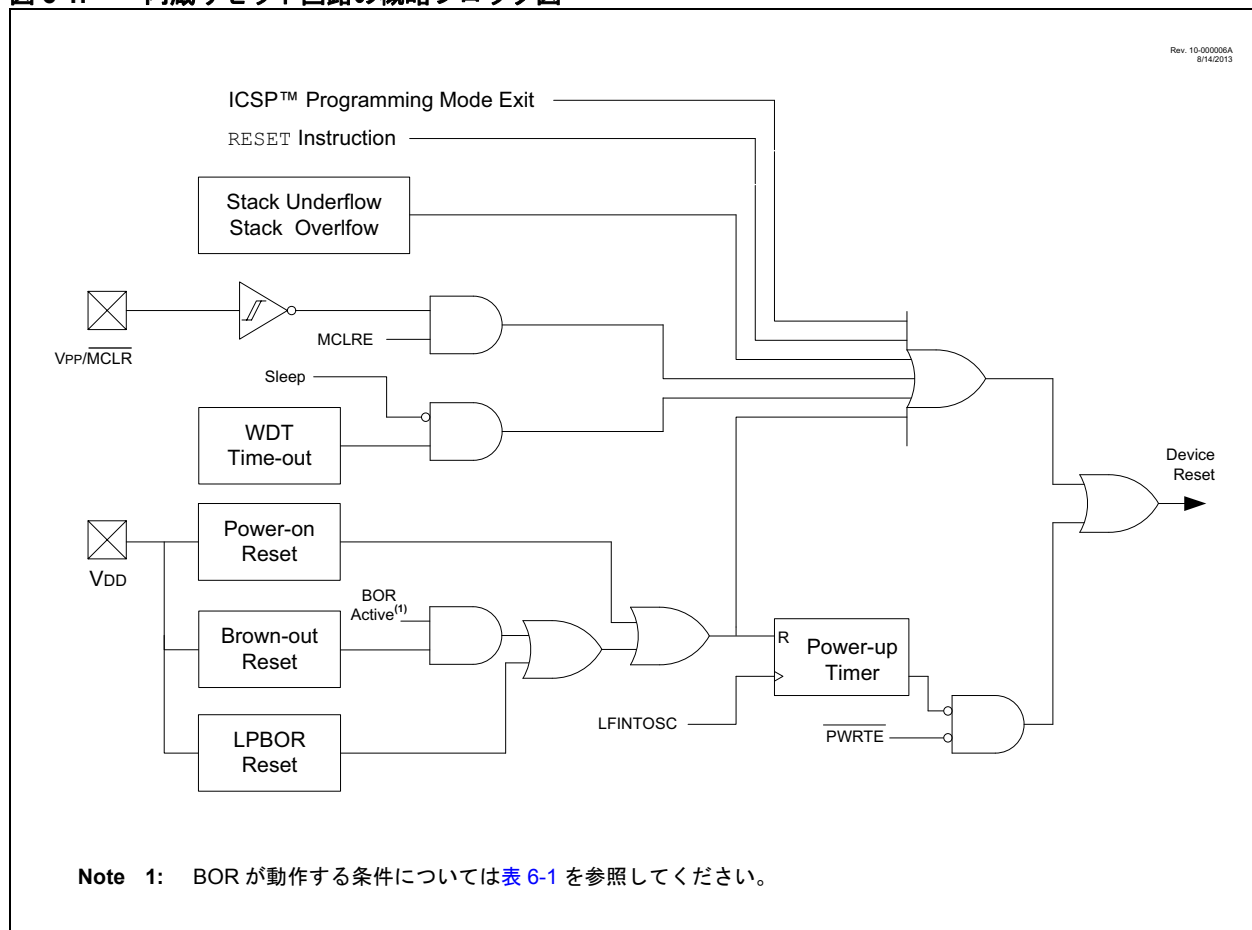
このデバイスは以下の方法でリセットできます。

- パワーオンリセット (POR)
- ブラウンアウトリセット (BOR)
- 低消費電力ブラウンアウトリセット (LPBOR)
- MCLR リセット
- WDT リセット
- RESET 命令
- スタック オーバーフロー
- スタック アンダーフロー
- プログラミング モードの終了

VDD を安定させるために、必要に応じて BOR または POR 後のリセット時間を延長するパワーアップ タイマを有効にする事もできます。

図 6-1 に、内蔵リセット回路の概略ブロック図を示します。

図 6-1: 内蔵リセット回路の概略ブロック図



PIC16(L)F1764/5/8/9

6.1 パワーオンリセット (POR)

VDD が最小許容値に達するまで、POR 回路はデバイスをリセット状態に保ちます。VDD の立ち上がりが遅い場合、デバイスを高速で動作させる場合、またはアナログ性能に制約がある場合、最小 VDD より高い設定が必要になることがあります。PWRT、BOR、MCLR 機能を使うと、デバイスの動作条件が全て満たされるまで起動期間を延長できます。

6.1.1 パワーアップ タイマ (PWRT)

パワーアップ タイマは、POR またはブラウンアウト リセット時に、公称 64 ms のタイムアウトを提供します。PWRT がアクティブの間、デバイスはリセット状態に維持されます。PWRT 遅延によって、VDD が許容レベルまで立ち上がる時間を余分に確保できます。パワーアップ タイマを有効にするには、コンフィグレーションワードの PWRT_E ビットをクリアします。

パワーアップ タイマは POR または BOR の解除後に起動します。

詳細は、アプリケーション ノート AN607 『Power-up Trouble Shooting』 (DS00607) を参照してください。

6.2 ブラウンアウト リセット (BOR)

BOR 回路は、VDD がユーザの設定した最低レベルまで低下すると、デバイスをリセット状態に維持します。POR と BOR を使う事で、デバイスが正常動作レンジ外の Vdd で動作する事を防止できます。

ブラウンアウト リセット モジュールにはコンフィグレーションワードの BOREN<1:0> ビットによって制御される、以下の 4 つの動作モードがあります。

- BOR を常時 ON にする
- BOR をスリープ中に OFF にする
- BOR をソフトウェアで制御する
- BOR を常時 OFF にする

詳細は表 6-1 を参照してください。

ブラウンアウトリセットをトリガする電圧値は、コンフィグレーションワードの BORV ビットで選択します。

VDD のノイズ除去フィルタは、小さな電圧変動によって BOR がトリガされる事を防ぎます。VDD がパラメータ TBORDC よりも長い期間 VBOR を下回ると、デバイスがリセットされます。詳細は図 6-2 を参照してください。

表 6-1: BOR の動作モード

BOREN<1:0>	SBOREN	デバイスモード	BOR モード	POR 解除時 / スリープからの復帰時の命令実行
11	X	X	アクティブ	BOR レディ (BORRDY = 1) を待機 ⁽¹⁾
10	X	通常動作	アクティブ	BOR レディ (BORRDY = 1) を待機
		スリープ	無効	
01	1	X	アクティブ	BOR レディ (BORRDY = 1) を待機 ⁽¹⁾
	0	X	無効	ただちに開始 (BORRDY = x)
00	X	X	無効	

Note 1: 「POR の解除」または「スリープからの復帰」という、これらの特定のケースでは、起動時に遅延が発生しません。CPU が命令を実行できるようになる前に、BOR レディフラグがセットされます (BORRDY = 1)。BOREN<1:0> ビットによって BOR 回路が強制的に ON になるためです。

6.2.1 BOR を常時 ON にする

コンフィグレーションワードの BOREN ビットを「11」にプログラムすると、BOR が常時 ON 状態を保ちます。BOR がレディ状態になり、VDD が BOR しきい値より高くなるまで、デバイスの起動を遅延させます。

BOR 保護はスリープ中も有効です。BOR によってスリープからの復帰時に遅延が発生する事はありません。

6.2.2 BOR をスリープ中に OFF にする

コンフィグレーションワードの BOREN ビットを「10」にプログラムすると、スリープ中を除いて BOR が ON 状態を保ちます。BOR がレディ状態になり、VDD が BOR しきい値より高くなるまで、デバイスの起動を遅延させます。

BOR 保護はスリープ中は無効です。スリープからの復帰時、BOR がレディ状態になるまで待機します。

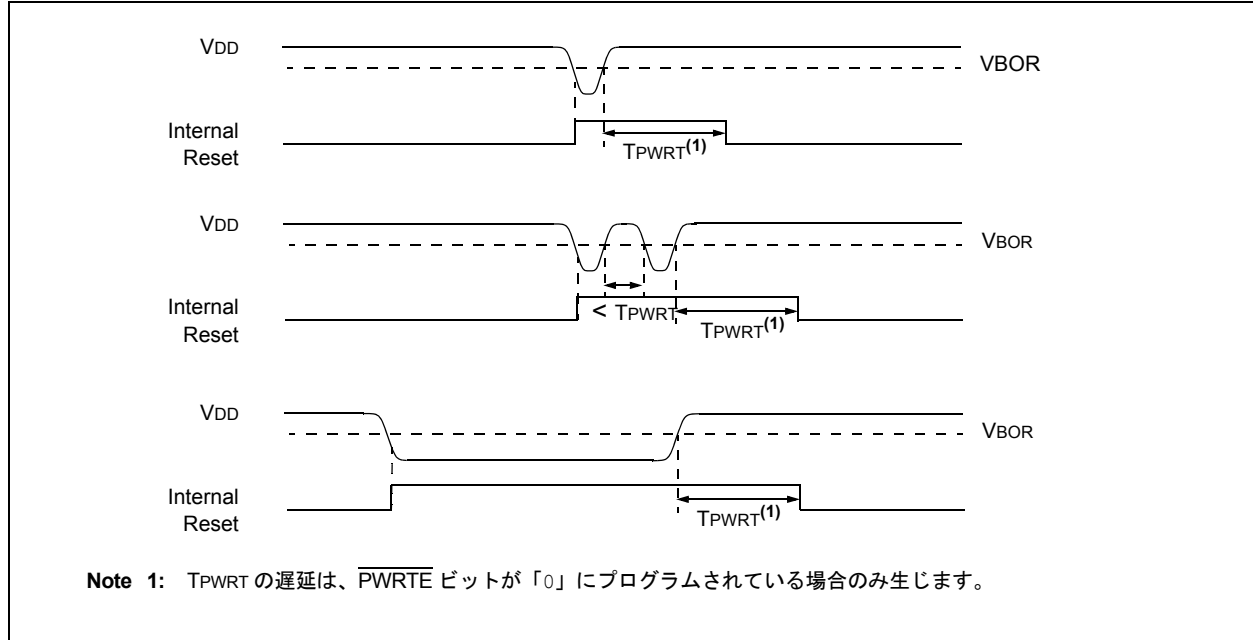
6.2.3 BOR をソフトウェアで制御する

コンフィグレーションワードの BOREN ビットを「01」にプログラムした場合の BOR の動作は BORCON レジスタの SBOREN ビットによって決まります。デバイスの起動が、BOR のレディ状態または VDD レベルによって遅延する事はありません。

BOR 保護は、BOR 回路の準備が整うとただちに開始します。BOR 回路の状態は、BORCON レジスタの BORRDY ビットに反映されます。

BOR 保護の状態はスリープによって変化しません。

図 6-2: ブラウンアウト挙動



6.3 レジスタ定義 :BOR 制御

レジスタ 6-1: BORCON: ブラウンアウト リセット制御レジスタ

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS ⁽¹⁾	—	—	—	—	—	BORRDY
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7 **SBOREN:** ソフトウェア制御ブラウンアウト リセット イネーブルビット

コンフィグレーションワードの BOREN<1:0> ≠ 01 の場合:

SBOREN は読み書き可能だが、BOR へ影響しない

コンフィグレーションワードの BOREN<1:0> = 01 の場合:

1 = BOR を有効にする

0 = BOR を無効にする

bit 6 **BORFS:** ブラウンアウト リセット高速起動ビット⁽¹⁾

BOREN<1:0> = 11 (常時 ON)、または BOREN<1:0> = 00 (常時 OFF) の場合:

BORFS は読み書き可能だが、BOR へ影響しない

BOREN<1:0> = 10 (スリープ中無効)、または BOREN<1:0> = 01 (ソフトウェア制御) の場合:

1 = バンドギャップを強制的に常時 ON にする (スリープ / 復帰 / 通常動作の各条件に適用)

0 = バンドギャップは通常動作するが、停止可能である

bit 5-1 **未実装:** 「0」として読み出し

bit 0 **BORRDY:** ブラウンアウト リセット回路レディステータスビット

1 = ブラウンアウト リセット回路がアクティブである

0 = ブラウンアウト リセット回路がアクティブでない

Note 1: BOREN<1:0> ビットはコンフィグレーションワード内にあります。

PIC16(L)F1764/5/8/9

6.4 低消費電力ブラウンアウト リセット (LPBOR)

低消費電力ブラウンアウト リセット (LPBOR) は、リセット サブシステムに不可欠な部分です。BOR と他のモジュールとの相互作用については、[図 6-1](#) を参照してください。

LPBOR は外部 VDD ピンの監視に使用します。一定の値より低い電圧が検出されると、デバイスはリセット状態に保持されます。その場合、レジスタビット (BOR) が変化し、BOR リセットが発生した事を示します。BOR と LPBOR は、どちらも同じビットをセットします。[レジスタ 6-2](#) を参照してください。

6.4.1 LPBOR の有効化

LPBOR はコンフィグレーションワードの LPBOR ビットで制御します。デバイスを消去すると、LPBOR モジュールは既定値 (無効) に戻ります。

6.4.1.1 LPBOR モジュールの出力

LPBOR モジュールの出力は、リセットをアサートするかどうかを示す信号です。この信号と BOR モジュールのリセット信号の論理和 (OR) を取る事で、PCON レジスタと電源制御ブロックに入力される汎用 BOR 信号が得られます。

6.5 MCLR

MCLR はデバイスをリセットする任意の外部入力です。MCLR 機能は、コンフィグレーションワードの MCLRE ビットとコンフィグレーションワードの LVP ビットで設定します ([表 6-2](#))。

表 6-2: MCLR の設定

MCLRE	LVP	MCLR
0	0	無効
1	0	有効
x	1	有効

6.5.1 MCLR が有効

MCLR が有効でこのピンが Low の場合、デバイスはリセット状態を維持します。MCLR ピンは、デバイス内部の弱プルアップ回路を介して VDD に接続されています。デバイスの MCLR リセット経路はノイズフィルタを備えています。このフィルタによって、小さなパルスは無視されます。

Note: リセットは MCLR ピンを Low に駆動しません。

6.5.2 MCLR が無効

MCLR が無効の場合、このピンは汎用入力として機能し、内部の弱プルアップ回路はソフトウェアが制御します。詳細は[セクション 11.1 「PORTA レジスタ」](#)を参照してください。

6.6 ウォッチドッグ タイマ (WDT) リセット

ウォッチドッグ タイマは、タイムアウト期間内にファームウェアが CLRWDT 命令を発行しなかった場合、リセットを生成します。STATUS レジスタの TO ビットと PD ビットの変化が WDT リセットを示します。詳細は[セクション 9.0 「ウォッチドッグ タイマ \(WDT\)」](#)を参照してください。

6.7 RESET 命令

RESET 命令はデバイスをリセットします。PCON レジスタの RI ビットが「0」に設定されます。RESET 命令実行後の既定値は[表 6-4](#)を参照してください。

6.8 スタック オーバーフロー/アンダーフロー リセット

スタックのオーバーフロー/アンダーフローが発生した場合にデバイスをリセットできます。PCON レジスタの STKOVF または STKUNF ビットによってリセット条件が示されます。これらのリセットは、コンフィグレーションワードの STVREN ビットをセットする事で有効にします。詳細は[セクション 3.6.2 「オーバーフロー/アンダーフローによるリセット」](#)を参照してください。

6.9 プログラミング モードの終了

プログラミング モードを終了すると、デバイスは POR 発生時のようにふるまいます。

6.10 パワーアップ タイマ

必要に応じてパワーアップ タイマを使い、BOR または POR イベント後のデバイス実行を遅延させる事ができます。このタイマは通常、デバイスが動作を開始する前に VDD を安定化させるために使います。

パワーアップ タイマはコンフィグレーションワードの PWRT \overline{E} ビットで制御します。

6.11 起動シーケンス

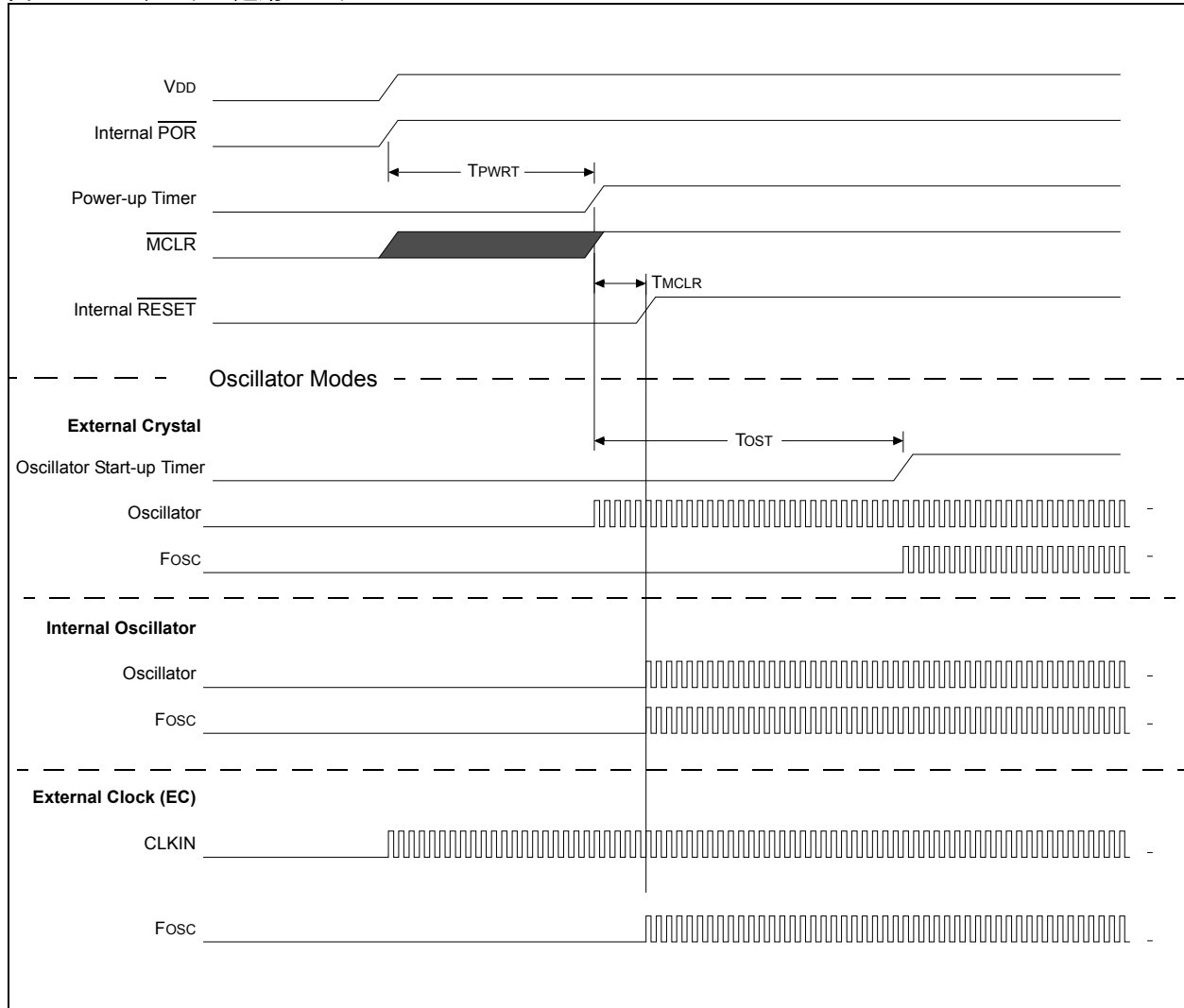
POR または BOR の解除後にデバイスが実行を開始するには、以下の条件が満たされる必要があります。

1. パワーアップ タイマのカウント完了 (有効な場合)
2. オシレータ起動タイマのカウント完了 (オシレータ源が必要としている場合)
3. MCLR の解除 (有効な場合)

タイムアウトの合計時間は、オシレータとパワーアップ タイマの設定で決まります。詳細は[セクション 5.0 「オシレータ モジュール \(フェイルセーフ クロック モニタ機能付き\)」](#)を参照してください。

パワーアップ タイマとオシレータ起動タイマは、MCLR リセットから独立して動作します。MCLR を Low に保持したまましていると、パワーアップ タイマとオシレータ起動タイマがタイムアウトします。この場合、MCLR を High にすると、10 Fosc サイクル後にプログラムの実行が開始します ([図 6-3](#) 参照)。この機能は、テスト目的や並列動作している複数デバイスの同期に使うと便利です。

図 6-3: リセット起動シーケンス



PIC16(L)F1764/5/8/9

6.12 リセット原因の特定

リセットが発生すると STATUS および PCON レジスタのビット値が変化します。これらのビットを観察するとリセットの原因が分かります。表 6-3 と表 6-4 にこれらのレジスタのリセット条件を示します。

表 6-3: リセットステータスビットとその意味

STKOVF	STKUNF	RWDT	RMCLR	RI	POR	BOR	TO	PD	条件
0	0	1	1	1	0	x	1	1	パワーオン リセット
0	0	1	1	1	0	x	0	x	不正、 \overline{TO} が \overline{POR} でセット
0	0	1	1	1	0	x	x	0	不正、 \overline{PD} が \overline{POR} でセット
0	0	u	1	1	u	0	1	1	ブラウンアウト リセット
u	u	0	u	u	u	u	0	u	WDT リセット
u	u	u	u	u	u	u	0	0	スリープからの WDT 復帰
u	u	u	u	u	u	u	1	0	スリープからの割り込み復帰
u	u	u	0	u	u	u	u	u	通常動作中の \overline{MCLR} リセット
u	u	u	0	u	u	u	1	0	スリープ中の \overline{MCLR} リセット
u	u	u	u	0	u	u	u	u	RESET 命令の実行
1	u	u	u	u	u	u	u	u	スタック オーバーフロー リセット (STVREN = 1)
u	1	u	u	u	u	u	u	u	スタック アンダーフロー リセット (STVREN = 1)

表 6-4: 特殊レジスタのリセット条件

条件	プログラムカウンタ	STATUS レジスタ	PCON レジスタ
パワーオン リセット	0000h	---1 1000	00-- 110x
通常動作中の \overline{MCLR} リセット	0000h	---u uuuu	uu-- 0uuu
スリープ中の \overline{MCLR} リセット	0000h	---1 0uuu	uu-- 0uuu
WDT リセット	0000h	---0 uuuu	uu-- uuuu
スリープからの WDT 復帰	PC + 1	---0 0uuu	uu-- uuuu
ブラウンアウト リセット	0000h	---1 1uuu	00-- 11u0
スリープからの割り込み復帰	PC + 1 ⁽¹⁾	---1 0uuu	uu-- uuuu
RESET 命令の実行	0000h	---u uuuu	uu-- u0uu
スタック オーバーフロー リセット (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
スタック アンダーフロー リセット (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し

Note 1: 割り込みによる復帰で、かつ GIE (グローバル割り込みイネーブルビット) がセットされている場合、リターンアドレス値がスタックにプッシュされ、PC+1 の実行後に PC に割り込みベクタ (0004h) が書き込まれます。

6.13 電源制御 (PCON) レジスタ

レジスタ 6-2 に、PCON レジスタの各ビットを示します。

電源制御 (PCON) レジスタには、以下のリセットの種類を見分けるためのフラグビットがあります。

- パワーオン リセット ($\overline{\text{POR}}$)
- ブラウンアウト リセット ($\overline{\text{BOR}}$)
- RESET 命令によるリセット ($\overline{\text{RI}}$)
- MCLR リセット ($\overline{\text{RMCLR}}$)
- ウォッチドッグ タイマリセット ($\overline{\text{RWD\overline{T}}}$)
- スタック アンダーフロー リセット (STKUNF)
- スタック オーバーフロー リセット (STKOVF)

6.14 レジスタ定義：電源制御

レジスタ 6-2: PCON: 電源制御レジスタ

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	$\overline{\text{RWD\overline{T}}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7						bit 0	

凡例:

HC = ビットはハードウェアでクリア

HS = ビットはハードウェアでセット

R = 読み出し可能ビット

W = 書き込み可能ビット

U = 未実装ビット、「0」として読み出し

u = ビットは不変

x = ビットは未知

-m/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット

「0」= ビットはクリア

q = 条件により異なる

- bit 7 **STKOVF:** スタック オーバーフロー フラグビット
 1 = スタック オーバーフローが発生した
 0 = スタック オーバーフローは発生していない、またはファームウェアによってクリアされた
- bit 6 **STKUNF:** スタック アンダーフロー フラグビット
 1 = スタック アンダーフローが発生した
 0 = スタック アンダーフローは発生していない、またはファームウェアによってクリアされた
- bit 5 **未実装:** 「0」として読み出し
- bit 4 **RWD $\overline{\text{T}}$:** ウォッチドッグ タイマリセット フラグビット
 1 = ウォッチドッグ タイマリセットは発生していない、またはファームウェアによって「1」にセットされた
 0 = ウォッチドッグ タイマリセットが発生した (ハードウェアによってクリアされた)
- bit 3 **RMCLR:** MCLR リセット フラグビット
 1 = $\overline{\text{MCLR}}$ リセットは発生していない、またはファームウェアによって「1」にセットされた
 0 = $\overline{\text{MCLR}}$ リセットが発生した (ハードウェアによってクリアされた)
- bit 2 **RI:** RESET 命令フラグビット
 1 = RESET 命令は実行されていない、またはファームウェアによって「1」に設定された
 0 = RESET 命令が実行された (ハードウェアによってクリアされた)
- bit 1 **POR:** パワーオン リセット ステータスビット
 1 = パワーオン リセットは発生していない
 0 = パワーオン リセットが発生した (パワーオン リセット発生後はソフトウェアでセットする必要がある)
- bit 0 **BOR:** ブラウンアウト リセット ステータスビット
 1 = ブラウンアウト リセットは発生していない
 0 = ブラウンアウト リセットが発生した (パワーオン リセットまたはブラウンアウト リセット発生後はソフトウェアでセットする必要がある)

PIC16(L)F1764/5/8/9

表 6-5: リセット関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ内容記載ページ
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	91
PCON	STKOVF	STKUNF	—	$\overline{RWD\overline{T}}$	\overline{RMCLR}	\overline{RI}	\overline{POR}	\overline{BOR}	95
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
WDTCON	—	—	WDTPS<4:0>					SWDTEN	117

凡例： — = 未実装、「0」として読み出し。網掛けの部分はリセットでは使いません。

7.0 割り込み

割り込み機能を使うと、特定のイベントが通常のプログラムフローに割り込むことができます。ファームウェアによって割り込み要因を判断し、それに基づいて動作します。MCU をスリープから復帰させるように設定できる割り込みもあります。

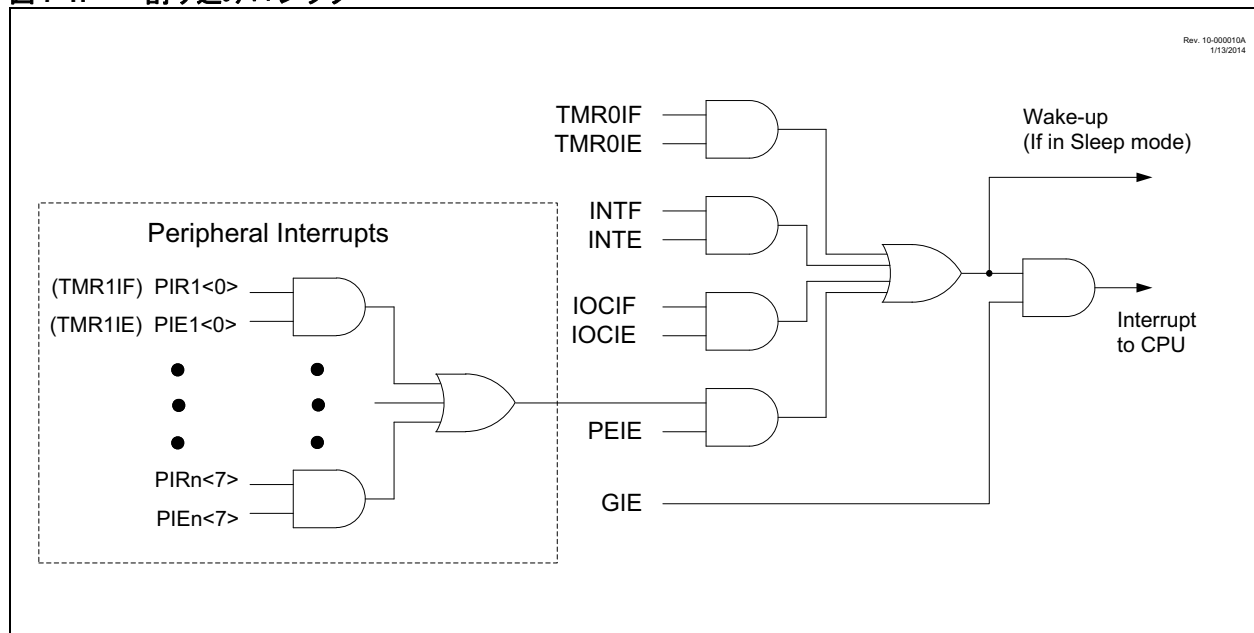
この章では、割り込みに関する以下の情報を提供します。

- 動作
- 割り込みレイテンシ
- スリープ中の割り込み
- INT ピン
- コンテキスト自動保存機能

多くの周辺モジュールが割り込みを生成します。詳細は、該当する章を参照してください。

図 7-1 に、割り込みロジックのブロック図を示します。

図 7-1: 割り込みロジック



PIC16(L)F1764/5/8/9

7.1 動作

デバイスリセット後、割り込みは無効です。有効にするには、以下のビットをセットします。

- INTCON レジスタの GIE ビット
- 特定の割り込みイベントに対する割り込みイネーブルビット
- INTCON レジスタの PEIE ビット (その割り込みイベントの割り込みイネーブルビットが PIE1 または PIE2 レジスタに含まれる場合)

INTCON、PIR1、PIR2 レジスタは、割り込みフラグビットにより各割り込みを記録します。割り込みフラグビットは、GIE、PEIE、各割り込みイネーブルビットの状態に関わらずセットされます。

GIE ビットがセットされている場合に割り込みイベントが発生すると、以下のイベントが発生します。

- 現在プリフェッチされている命令がフラッシュ (消去) される。
- GIE ビットがクリアされる。
- 現在の PC (プログラム カウンタ) がスタックにプッシュされる。
- 重要なレジスタ値がシャドウレジスタに自動保存される ([セクション 7.5 「コンテキスト自動保存機能」](#) 参照) 。
- PC に割り込みベクタ 0004h が書き込まれる。

割り込みサービスルーチン (ISR) のファームウェアでは、割り込みフラグビットをポーリングして割り込み要因を判断します。割り込み動作の繰り返しを避けるため、ISR から抜ける前に割り込みフラグビットをクリアする必要があります。ISR 実行中に発生する割り込みは全て割り込みフラグで記録されますが、GIE ビットがクリアされているためプロセッサがその割り込みベクタにリダイレクトする事はありません。

RETFIE 命令では、割り込み前に実行していたアドレスをスタックからポップし、シャドウレジスタに保存されていたコンテキストを復元し、GIE ビットをセットする事によって、ISR から通常動作に復帰します。

具体的な割り込み動作のその他の情報は、各周辺モジュールの章を参照してください。

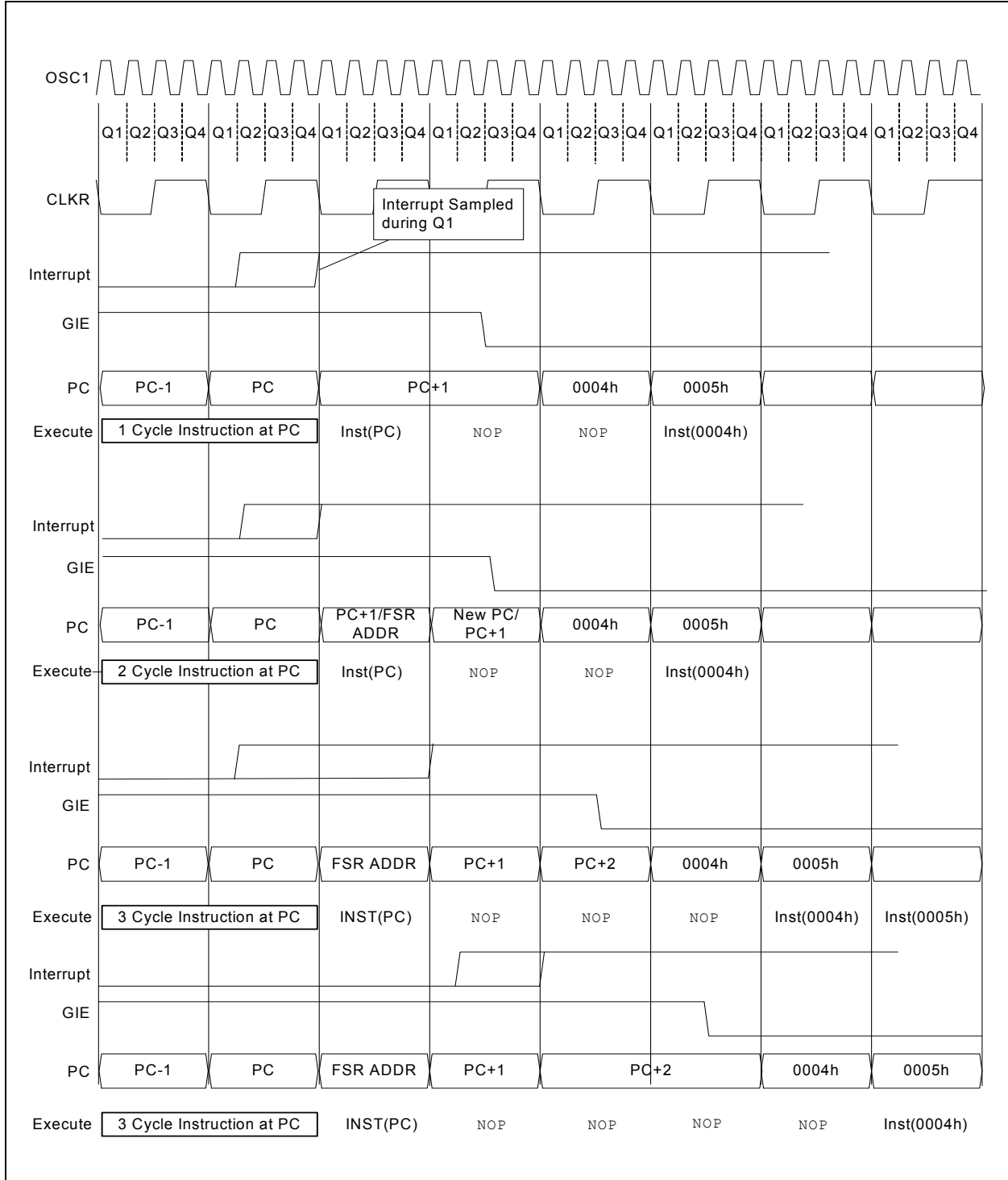
Note 1: 各割り込みフラグビットは、その他のイネーブルビットの状態とは無関係にセットされます。

2: GIE ビットがクリアされている限り、全ての割り込みは無視されます。GIE ビットがクリアされている間に発生した全ての割り込みは、GIE ビットが再びセットされた時点で処理されます。

7.2 割り込みレイテンシ

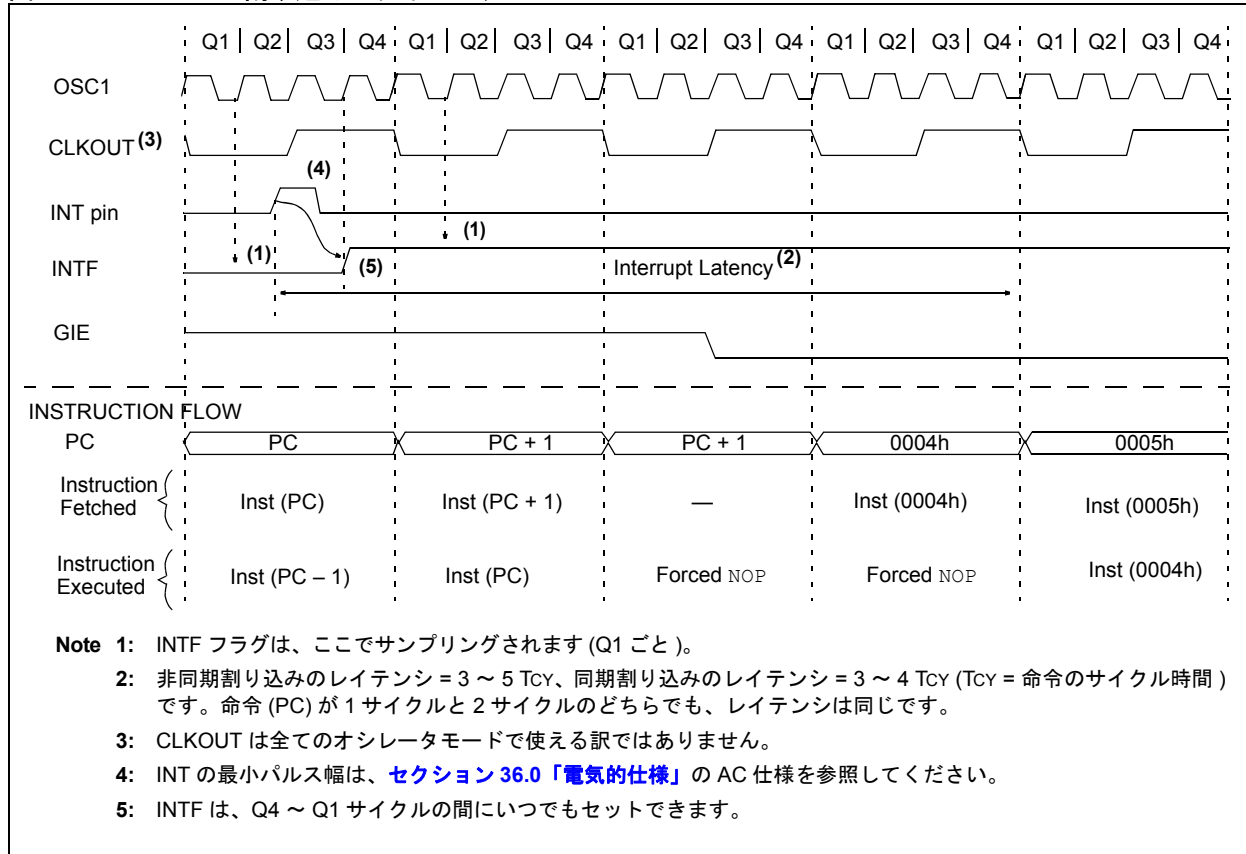
割り込みレイテンシとは、割り込みイベントが発生してから、割り込みベクタのコード実行が開始するまでの時間を意味します。同期割り込みのレイテンシは、3 ~ 4 命令サイクルです。非同期割り込みのレイテンシは割り込み発生タイミングによって異なり、3 ~ 5 命令サイクルです。詳細は [図 7-2](#) と [図 7-3](#) を参照してください。

図 7-2: 割り込みレイテンシ



PIC16(L)F1764/5/8/9

図 7-3: INT ピン割り込みのタイミング



7.3 スリープ中の割り込み

割り込みの種類によっては、スリープからの復帰に使えません。この場合、周辺モジュールがシステムクロックを使わずに動作できる必要があります。使う割り込み要因のイネーブルビットは、スリープに移行する前にセットしておく必要があります。

スリープから復帰する際、GIE ビットもセットされていると、プロセッサは割り込みベクタへ分岐します。セットされていない場合、SLEEP 命令の次の命令から実行を再開します。SLEEP 命令の直後の命令は、ISR へ分岐する前に必ず実行されます。詳細は[セクション 8.0「パワーダウンモード\(スリープ\)」](#)を参照してください。

7.4 INT ピン

INT ピンを使うと、非同期のエッジトリガ型割り込みを生成できます。INTCON レジスタの INTE ビットをセットすると、この割り込みが有効化されます。OPTION_REG レジスタの INTEDG ビットで割り込みを発生させるエッジを設定します。INTEDG ビットがセットされている場合、立ち上がりエッジで割り込みが発生します。INTEDG ビットがクリアされている場合、立ち下がりエッジで割り込みが発生します。INT ピンに有効なエッジが表れると、INTCON レジスタの INTF ビットがセットされます。この時 GIE ビットと INTE ビットもセットされていると、プロセッサはプログラムの実行を割り込みベクタへリダイレクトします。

7.5 コンテキスト自動保存機能

割り込み動作に移行する際、PC の戻りアドレス値がスタックに保存されます。また、以下に示すレジスタ値も自動的にシャドウレジスタに保存されます。

- W レジスタ
- STATUS レジスタ (\overline{TO} と \overline{PD} を除く)
- BSR レジスタ
- FSR レジスタ
- PCLATH レジスタ

割り込みサービスルーチンから通常動作に戻る際、これらのレジスタ値は自動的に復元されます。ISR 中にこれらのレジスタに加えられた変更は、全て失われます。変更が失われないようにするには、対応するシャドウレジスタも変更しておきます。これにより、ISR 終了時に変更後の値が書き戻されます。シャドウレジスタはバンク 31 内にあり、読み書き可能です。ユーザアプリケーションによっては、その他のレジスタ値の保存が必要になる場合があります。

PIC16(L)F1764/5/8/9

7.6 レジスタ定義：割り込み制御

レジスタ 7-1: INTCON: 割り込み制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF ⁽¹⁾
bit 7							bit 0

凡例：

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **GIE:** グローバル割り込みイネーブルビット
1 = 全てのアクティブな割り込みを有効にする
0 = 全ての割り込みを無効にする
- bit 6 **PEIE:** 周辺モジュール割り込みイネーブルビット
1 = 全てのアクティブな周辺モジュール割り込みを有効にする
0 = 全ての周辺モジュール割り込みを無効にする
- bit 5 **TMR0IE:** Timer0 オーバーフロー割り込みイネーブルビット
1 = Timer0 割り込みを有効にする
0 = Timer0 割り込みを無効にする
- bit 4 **INTE:** INT 外部割り込みイネーブルビット
1 = INT 外部割り込みを有効にする
0 = INT 外部割り込みを無効にする
- bit 3 **IOCFIE:** 状態変化割り込みイネーブルビット
1 = 状態変化割り込みを有効にする
0 = 状態変化割り込みを無効にする
- bit 2 **TMR0IF:** Timer0 オーバーフロー割り込みフラグビット
1 = TMR0 レジスタがオーバーフローした
0 = TMR0 レジスタはオーバーフローしていない
- bit 1 **INTF:** INT 外部割り込みフラグビット
1 = INT 外部割り込みが発生した
0 = INT 外部割り込みは発生していない
- bit 0 **IOCFIF:** 状態変化割り込みフラグビット⁽¹⁾
1 = 状態変化割り込みピンのうち少なくとも 1 本の状態が変化した
0 = 状態変化割り込みピンの状態は変化していない

Note 1: IOCFIF フラグビットは読み出し専用で、IOCFIF レジスタの全ての状態変化割り込みフラグがソフトウェアによってクリアされると、このビットもクリアされます。

Note: 割り込み条件が発生すると、対応するイネーブルビットまたはグローバルイネーブルビット (INTCON レジスタの GIE) の状態に関係なく割り込みフラグビットがセットされます。割り込みを有効にする前に、対応する割り込みフラグビットをユーザソフトウェアで確実にクリアしておく必要があります。

レジスタ 7-2: PIE1: 周辺モジュール割り込みイネーブル レジスタ 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **TMR1GIE:** Timer1 ゲート割り込みイネーブルビット
1 = Timer1 のゲート アクイジション割り込みを有効にする
0 = Timer1 のゲート アクイジション割り込みを無効にする
- bit 6 **ADIE:** A/D コンバータ (ADC) 割り込みイネーブルビット
1 = ADC 割り込みを有効にする
0 = ADC 割り込みを無効にする
- bit 5 **RCIE:** EUSART 受信割り込みイネーブルビット
1 = EUSART 受信割り込みを有効にする
0 = EUSART 受信割り込みを無効にする
- bit 4 **TXIE:** EUSART 送信割り込みイネーブルビット
1 = EUSART 送信割り込みを有効にする
0 = EUSART 送信割り込みを無効にする
- bit 3 **SSP1IE:** 同期シリアルポート (MSSP) 割り込みイネーブルビット
1 = MSSP 割り込みを有効にする
0 = MSSP 割り込みを無効にする
- bit 2 **CCP1IE:** CCP1 割り込みイネーブルビット
1 = CCP1 割り込みを有効にする
0 = CCP1 割り込みを無効にする
- bit 1 **TMR2IE:** TMR2/PR2 一致割り込みイネーブルビット
1 = Timer2/PR2 一致割り込みを有効にする
0 = Timer2/PR2 一致割り込みを無効にする
- bit 0 **TMR1IE:** Timer1 オーバーフロー割り込みイネーブルビット
1 = Timer1 オーバーフロー割り込みを有効にする
0 = Timer1 オーバーフロー割り込みを無効にする

Note: 周辺モジュール割り込みを有効にするには、INTCON レジスタの PEIE ビットをセットする必要があります。

PIC16(L)F1764/5/8/9

レジスタ 7-3: **PIE2: 周辺モジュール割り込みイネーブル レジスタ 2**

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **OSFIE:** オシレータ障害割り込みイネーブルビット
1 = オシレータ障害割り込みを有効にする
0 = オシレータ障害割り込みを無効にする
- bit 6 **C2IE:** コンパレータ C2 割り込みイネーブルビット
1 = コンパレータ C2 割り込みを有効にする
0 = コンパレータ C2 割り込みを無効にする
- bit 5 **C1IE:** コンパレータ C1 割り込みイネーブルビット
1 = コンパレータ C1 割り込みを有効にする
0 = コンパレータ C1 割り込みを無効にする
- bit 4 **未実装:** 「0」として読み出し
- bit 3 **BCL1IE:** MSSP バスコリジョン割り込みイネーブルビット
1 = MSSP バスコリジョン割り込みを有効にする
0 = MSSP バスコリジョン割り込みを無効にする
- bit 2 **C4IE:** TMR6/T6PR 一致割り込みイネーブルビット⁽¹⁾
1 = コンパレータ C4 割り込みを有効にする
0 = コンパレータ C4 割り込みを無効にする
- bit 1 **C3IE:** TMR4/T4PR 一致割り込みイネーブルビット⁽¹⁾
1 = コンパレータ C3 割り込みを有効にする
0 = コンパレータ C3 割り込みを無効にする
- bit 0 **CCP2IE:** CCP2 割り込みイネーブルビット⁽¹⁾
1 = CCP2 割り込みを有効にする
0 = CCP2 割り込みを無効にする

Note 1: PIC16(L)F1768/9 の場合のみです。

Note: 周辺モジュール割り込みを有効にするには、INTCON レジスタの PEIE ビットをセットする必要があります。

レジスタ 7-4: PIE3: 周辺モジュール割り込みイネーブル レジスタ 3

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **PWM6IE:** PWM6 割り込みイネーブルビット ⁽¹⁾
1 = PWM6 割り込みを有効にする
0 = PWM6 割り込みを無効にする
- bit 6 **PWM5IE:** PWM5 割り込みイネーブルビット
1 = PWM5 割り込みを有効にする
0 = PWM5 割り込みを無効にする
- bit 5 **COG1IE:** COG1 自動シャットダウン割り込みイネーブルビット
1 = COG1 割り込みを有効にする
0 = COG1 割り込みを無効にする
- bit 4 **ZCDIE:** ゼロクロス検出 (ZCD) 割り込みイネーブルビット
1 = ZCD 割り込みを有効にする
0 = ZCD 割り込みを無効にする
- bit 3 **COG2IE:** COG2 自動シャットダウン割り込みイネーブルビット ⁽¹⁾
1 = COG2 割り込みを有効にする
0 = COG2 割り込みを無効にする
- bit 2 **CLC3IE:** CLC3 割り込みイネーブルビット
1 = CLC3 割り込みを有効にする
0 = CLC3 割り込みを無効にする
- bit 1 **CLC2IE:** CLC2 割り込みイネーブルビット
1 = CLC2 割り込みを有効にする
0 = CLC2 割り込みを無効にする
- bit 0 **CLC1IE:** CLC1 割り込みイネーブルビット
1 = CLC1 割り込みを有効にする
0 = CLC1 割り込みを無効にする

Note 1: PIC16(L)F1768/9 の場合のみです。

Note: 周辺モジュール割り込みを有効にするには、INTCON レジスタの PEIE ビットをセットする必要があります。

PIC16(L)F1764/5/8/9

レジスタ 7-5: PIR1: 周辺モジュール割り込み要求レジスタ 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **TMR1GIF:** Timer1 ゲート割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 6 **ADIF:** A/D コンバータ (ADC) 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 5 **RCIF:** EUSART 受信割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 4 **TXIF:** EUSART 送信割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 3 **SSP1IF:** 同期シリアルポート (MSSP) 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 2 **CCP1IF:** CCP1 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 1 **TMR2IF:** Timer2/T2PR 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 0 **TMR1IF:** Timer1 オーバーフロー割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない

Note: 割り込み条件が発生すると、対応するイネーブルビットまたはグローバルイネーブルビット (INTCON レジスタの GIE) の状態に関係なく割り込みフラグビットがセットされます。割り込みを有効にする前に、対応する割り込みフラグビットをユーザソフトウェアで確実にクリアしておく必要があります。

レジスタ 7-6: PIR2: 周辺モジュール割り込み要求レジスタ 2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **OSFIF:** オシレータ障害割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 6 **C2IF:** コンパレータ C2 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 5 **C1IF:** コンパレータ C1 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 4 **未実装:** 「0」として読み出し
- bit 3 **BCL1IF:** MSSP バスコリジョン割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 2 **C4IF:** コンパレータ C4 割り込みフラグビット⁽¹⁾
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 1 **C3IF:** コンパレータ C3 割り込みフラグビット⁽¹⁾
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 0 **CCP2IF:** CCP2 割り込みフラグビット⁽¹⁾
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない

Note 1: PIC16(L)F1768/9 の場合のみです。

Note: 割り込み条件が発生すると、対応するイネーブルビットまたはグローバルイネーブルビット (INTCON レジスタの GIE) の状態に関係なく割り込みフラグビットがセットされます。割り込みを有効にする前に、対応する割り込みフラグビットをユーザソフトウェアで確実にクリアしておく必要があります。

PIC16(L)F1764/5/8/9

レジスタ 7-7: PIR3: 周辺モジュール割り込み要求レジスタ 3

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **PWM6IF:** PWM6 割り込みフラグビット⁽¹⁾
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 6 **PWM5IF:** PWM5 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 5 **COG1IF:** COG1 自動シャットダウン割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 4 **ZCDIF:** ゼロクロス検出割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 3 **COG2IF:** COG2 自動シャットダウン割り込みフラグビット⁽¹⁾
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 2 **CLC3IF:** CLC3 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 1 **CLC2IF:** CLC2 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない
- bit 0 **CLC1IF:** CLC1 割り込みフラグビット
1 = 割り込みを保留中である
0 = 割り込みを保留中ではない

Note 1: PIC16(L)F1768/9 の場合のみです。

Note: 割り込み条件が発生すると、対応するイネーブルビットまたはグローバルイネーブルビット (INTCON レジスタの GIE) の状態に関係なく割り込みフラグビットがセットされます。割り込みを有効にする前に、対応する割り込みフラグビットをユーザソフトウェアで確実にクリアしておく必要があります。

表 7-1: 割り込み関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCF	102
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108

凡例： — = 未実装、「0」として読み出し。網掛けの部分は割り込みでは使いません。

Note 1: PIC16(L)F1768/9のみです。

8.0 パワーダウンモード(スリープ)

SLEEP 命令を実行すると、パワーダウンモードに移行します。

スリープに移行したデバイスの状態は、以下の通りです。

1. WDT がクリアされる。スリープ中の動作を有効に設定している場合、値はクリアされてもカウンタは継続する。
2. STATUS レジスタの $\overline{\text{PD}}$ ビットがクリアされる。
3. STATUS レジスタの $\overline{\text{TO}}$ ビットがセットされる。
4. CPU クロックが無効化される。
5. 31 kHz LFINTOSC は影響を受けない。このオシレータで動作する周辺モジュールは、スリープ中も動作を継続できる。
6. Timer1 のクロック源として以下を選択した場合、Timer1 自体と Timer1 で動作する周辺モジュールはスリープ中も動作を継続する。
 - LFINTOSC
 - T1CKI
 - セカンダリ オシレータ
7. ADC は影響を受けない。ただし専用の FRC オシレータが選択されている場合に限る。
8. I/O ポートは SLEEP 命令実行前の状態 (High、Low、ハイインピーダンスのどれか) を維持する。
9. WDT 以外のリセットはスリープの影響を受けない。

スリープ中の周辺モジュールの動作に関する詳細は、各周辺モジュールの章を参照してください。

消費電流を最小限に抑えるために、以下の条件に注意します。

- I/O ピンをフローティングにしない
- I/O ピンから電流をシンクする外部回路
- I/O ピンから電流をソースする内部回路
- 内部弱プルアップ回路によるピンからの電流流入
- 31 kHz LFINTOSC を使うモジュール
- セカンダリ オシレータを使うモジュール

ハイインピーダンス入力の I/O ピンは、外部で VDD または Vss に接続して、フローティング入力によるスイッチング電流が流れないようにします。

電流をソースする可能性がある内部回路の例として、DAC や FVR モジュールが挙げられます。これらのモジュールの詳細は、

[セクション 17.0 「5 ビット D/A コンバータ \(DAC\) モジュール」](#)と[セクション 14.0 「固定参照電圧 \(FVR\)」](#)を参照してください。

8.1 スリープからの復帰

以下のイベントのどれかによってスリープから復帰できます。

1. $\overline{\text{MCLR}}$ ピンへの外部リセット入力 (有効な場合)
2. BOR リセット (有効な場合)
3. POR リセット
4. ウォッチドッグ タイマ (有効な場合)
5. 全ての外部割り込み
6. スリープ中も動作可能な周辺モジュールによる割り込み (詳細は各周辺モジュールの章参照)

上記の 1 ~ 3 ではデバイスがリセットされます。4 ~ 6 は、プログラム実行の継続と見なされます。デバイスリセットまたは復帰イベントのどちらが発生したかを判断する方法は、[セクション 6.12 「リセット原因の特定」](#)を参照してください。

SLEEP 命令の実行中、次の命令 (PC+1) がプリフェッチされます。割り込みイベントでデバイスを復帰させるには、対応する割り込みイネーブルビットを有効にしておく必要があります。復帰は、GIE ビットの状態に関係なく実行されます。GIE ビットが無効の場合、デバイスは SLEEP 命令の次の命令から実行を再開します。GIE ビットが有効の場合、デバイスは SLEEP 命令の次の命令を実行後、割り込みサービスルーチン (ISR) を呼び出します。SLEEP 命令直後の命令を実行しない場合、SLEEP 命令の後に NOP 命令を配置します。

デバイスがスリープから復帰すると、復帰要因に関係なく WDT はクリアされます。

8.1.1 割り込みによる復帰

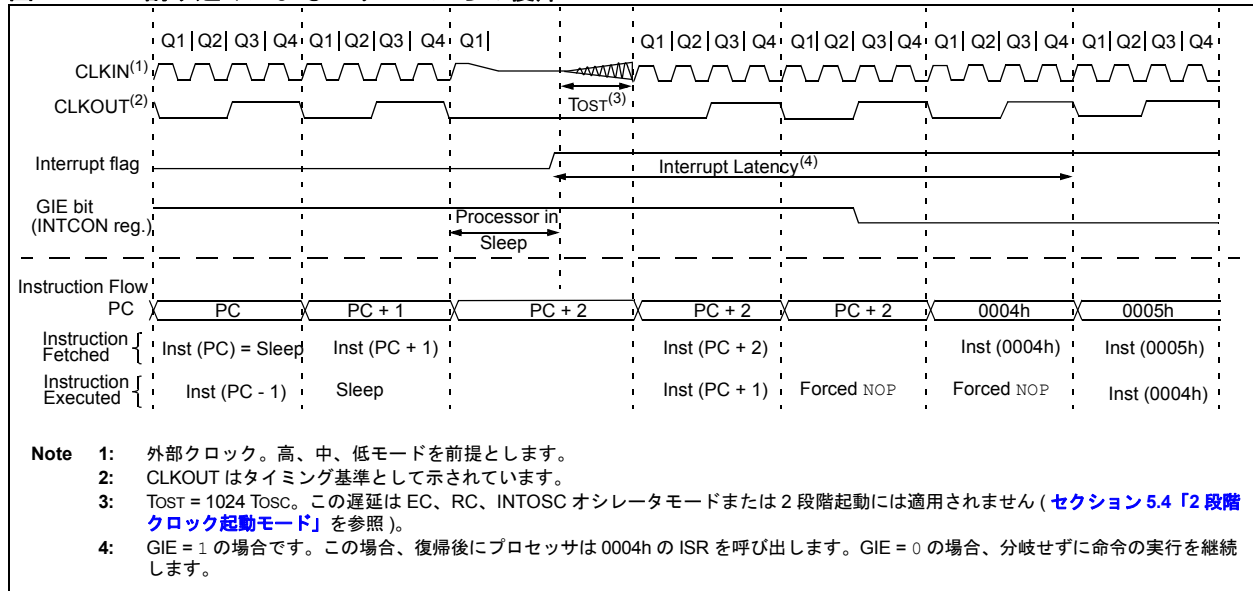
グローバル割り込みが無効 (GIE がクリア) で、何らかの割り込み要因の割り込みイネーブルビットと割り込みフラグビットの両方がセットされると、以下のどれかが発生します。

- SLEEP 命令の**実行前**に割り込みが発生した場合
 - SLEEP 命令が NOP として実行される。
 - WDT と WDT プリスケータはクリアされない。
 - STATUS レジスタの \overline{TO} ビットはセットされない。
 - STATUS レジスタの \overline{PD} ビットはクリアされない。

- SLEEP 命令の**実行中または実行後**に割り込みが発生した場合
 - SLEEP 命令が最後まで実行される。
 - デバイスがただちにスリープから復帰する。
 - WDT と WDT プリスケータがクリアされる。
 - STATUS レジスタの \overline{TO} ビットがセットされる。
 - STATUS レジスタの \overline{PD} ビットがクリアされる。

SLEEP 命令の実行前にフラグビットをチェックしたとしても、その後 SLEEP 命令が完了するまでにフラグビットがセットされる可能性があります。SLEEP 命令が実行されたかどうかを判断するには、 \overline{PD} ビットを検証します。 \overline{PD} ビットがセットされている場合、SLEEP 命令が NOP として実行された事を示します。

図 8-1: 割り込みによるスリープからの復帰



8.2 低消費電力スリープ

PIC16F1764/5/8/9 は、低ドロップアウト (LDO) 電圧レギュレータを内蔵しています。このため、I/O ピンを 5.5 V で動作させながら、内部デバイスロジックをこれより低い電圧で動作させる事ができます。LDO と関連基準回路は、デバイスがスリープ中でも動作を継続する必要があります。PIC16F1764/5/8/9 では、スリープ中の動作電流を、アプリケーション要件に応じてユーザが最適化できます。

低消費電力スリープを選択するには、VREGCON レジスタの VREGPM ビットをセットします。このビットをセットすると、デバイスのスリープ中に LDO と基準回路が低消費電力状態に移行します。

8.2.1 スリープ電流と復帰時間

既定値の動作モードでは、LDO と基準回路はスリープ中でも通常動作を継続します。全ての回路がスリープ中でも通常動作を継続するため、デバイスは素早くスリープから復帰できます。これに対して、低消費電力スリープから復帰する場合、これらの回路が通常動作を再開して安定化するためには一定の時間が必要です。

低消費電力スリープは、スリープ時間の長いアプリケーションに適しています。通常のスリープは、スリープから迅速かつ頻繁に復帰する必要があるアプリケーションに適しています。

8.2.2 スリープ中の周辺モジュールの使用

周辺モジュールによっては、通常のスリープ中なら動作可能でも、低消費電力スリープ中には正常に動作しないものがあります。低消費電力スリープは、以下のモジュールのみと一緒に使う事を前提としています。

- ブラウンアウト リセット (BOR)
- ウォッチドッグ タイマ (WDT)
- 外部割り込みピン / 状態変化割り込みピン
- Timer1 (100 kHz 未満の外部クロック源を使用)

Note: PIC16LF1764/5/8/9 では、通常のスリープと低消費電力スリープを選ぶ事はできません。PIC16LF1764/5/8/9 のスリープは常に低消費電力スリープであり、しかも復帰時に遅延が発生しません。これらのデバイスの最大 V_{DD} と I/O 電圧は、PIC16F1764/5/8/9 よりも低く設定されています。
詳細は [セクション 36.0「電気的仕様」](#) を参照してください。

8.3 レジスタ定義：電圧レギュレータ制御

レジスタ 8-1: VREGCON: 電圧レギュレータ制御レジスタ (1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	
—	—	—	—	—	—	VREGPM	予約済み	
bit 7							bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット 「0」= ビットはクリア

bit 7-2 **未実装:** 「0」として読み出し

bit 1 **VREGPM:** 電圧レギュレータ電力モード選択ビット

- 1 = 低消費電力スリープを有効にする (2)
スリープ中の消費電流は小さいが、復帰に時間を要する
- 0 = 通常消費電力モードを有効にする (2)
スリープ中の消費電流は大きい、短時間で復帰できる

bit 0 **予約済み:** 「1」として読み出し。このビットはセットしたままにします。

Note 1: PIC16F1764/5/8/9 の場合のみです。

2: [セクション 36.0「電氣的仕様」](#) を参照してください。

表 8-1: パワーダウン モード関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	166
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	166
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	167
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	167
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	168
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	168
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	169
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	169
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	169
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
VREGCON ⁽²⁾	—	—	—	—	—	—	VREGPM	予約済み	113
WDTCON	—	—	WDTPS<4:0>					SWDTEN	117

凡例: — = 未実装、「0」として読み出し。網掛けの部分はパワーダウン モードでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

2: PIC16F1764/5/8/9 のみです。

PIC16(L)F1764/5/8/9

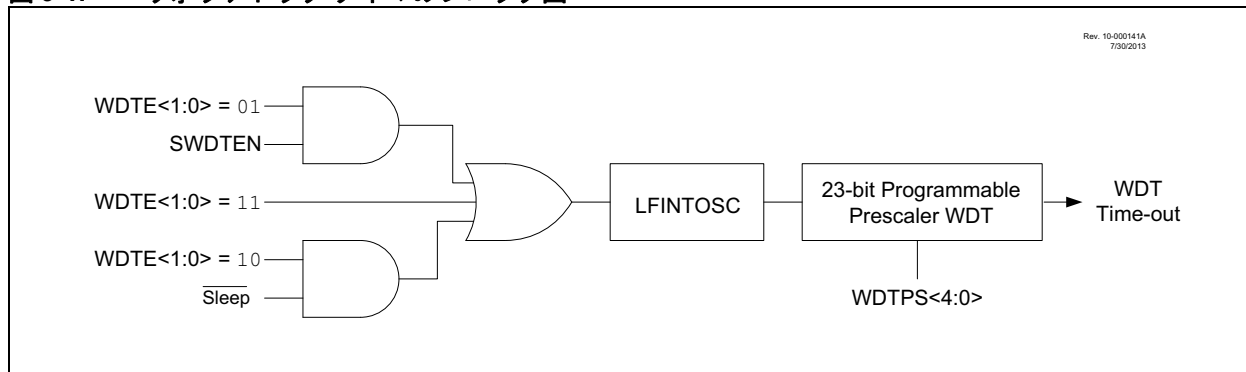
9.0 ウォッチドッグ タイマ (WDT)

ウォッチドッグ タイマは、一定時間が経過してもファームウェアが CLRWDT 命令を発行しないとリセットを発生させるシステムタイマです。通常、ウォッチドッグ タイマは予期しないイベントからシステムを回復する目的で使います。

WDT には以下の特長があります。

- 独立したクロック源
- 複数の動作モード
 - WDT を常時 ON にする
 - スリープ中は WDT を OFF にする
 - WDT をソフトウェアで制御する
 - WDT を常時 OFF にする
- タイムアウトを 1 ms ~ 256 s (公称) の範囲で設定可能
- 複数のリセット条件
- スリープ中でも動作

図 9-1: ウォッチドッグ タイマのブロック図



9.1 独立したクロック源

WDT は、31 kHz の LFINTOSC 内部オシレータで動作します。この章では、時間間隔を公称値 1 ms として説明します。LFINTOSC の仕様は、[表 36-8: オシレータのパラメータ](#)を参照してください。

9.2 WDT の動作モード

ウォッチドッグタイマモジュールには4つの動作モードがあり、コンフィグレーションワードのWDTE<1:0>ビットで設定します。[表 9-1](#)を参照してください。

9.2.1 WDT を常時 ON にする

コンフィグレーションワードのWDTEビットを「11」に設定すると、WDT が常時 ON 状態を保ちます。

WDT 保護はスリープ中も有効です。

9.2.2 スリープ中は WDT を OFF にする

コンフィグレーションワードのWDTEビットを「10」に設定すると、スリープ中はWDTがOFFになります。

WDT 保護はスリープ中には無効です。

9.2.3 WDT をソフトウェアで制御する

コンフィグレーションワードのWDTEビットを「01」に設定すると、WDTがWDTCONレジスタのSWDTENビットによって制御されます。

WDT 保護の状態はスリープに移行しても変化しません。詳細は[表 9-1](#)を参照してください。

表 9-1: WDT の動作モード

WDTE<1:0>	SWDTEN	デバイスモード	WDT のモード
11	X	X	アクティブ
10	X	通常動作	アクティブ
		スリープ	無効
01	1	X	アクティブ
	0		無効
00	X	X	無効

9.3 タイムアウト

WDTCON レジスタのWDTPSビットで、タイムアウトを 1 ms ~ 256 s (公称) の範囲で設定します。リセット後のタイムアウト既定値は 2 s です。

9.4 WDT のクリア

WDT は、以下のいずれかの場合にクリアされます。

- 何らかのリセットが発生した場合
- CLRWDT 命令を実行した場合
- デバイスがスリープに移行した場合
- デバイスがスリープから復帰した場合
- オシレータに障害が発生した場合
- WDT を無効にした場合
- オシレータ起動タイマ (OST) が動作中の場合

詳細は[表 9-2](#)を参照してください。

9.5 スリープ中の動作

デバイスがスリープに移行する時点で、WDT はクリアされます。スリープ中も WDT を有効にしている場合、WDT はカウントを再開します。

デバイスがスリープから復帰する時点で、WDT は再びクリアされます。OST が有効な場合、OST がタイムアウトするまでWDTはクリアされたままです。OSTの詳細は、[セクション 5.0「オシレータモジュール \(フェイルセーフクロックモニタ機能付き\)」](#)を参照してください。

デバイスがスリープ中にWDTがタイムアウトしても、リセットは発生しません。この場合、デバイスは復帰して動作を再開します。STATUSレジスタのTOビットとPDビットが変化してこのイベントの発生を示します。詳細は、STATUSレジスタの説明 ([レジスタ 3-1](#))を参照してください。

PIC16(L)F1764/5/8/9

表 9-2: WDT クリアの条件

条件	WDT
WDTE<1:0> = 00	クリア
WDTE<1:0> = 01 かつ SWDTEN = 0	
WDTE<1:0> = 10 でスリープに移行	
CLRWDT コマンド	
オシレータの障害を検出	
スリープからの復帰 (システムクロックが T1OSC、EXTRC、INTOSC、EXTCLK の場合)	
スリープからの復帰 (システムクロックが XT、HS、LP の場合)	OST タイムアウトまでクリアのまま
INTOSC 分周比 (IRCF ビット) を変更	そのまま

9.6 レジスタ定義：ウォッチドッグ制御

レジスタ 9-1: WDTCON: ウォッチドッグ タイマ制御レジスタ

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0	
—	—	WDTPS<4:0> ⁽¹⁾					SWDTEN	
bit 7							bit 0	

凡例：

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-m/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-6 **未実装**：「0」として読み出し

bit 5-1 **WDTPS<4:0>**: ウォッチドッグ タイマ時間選択ビット⁽¹⁾

ビット値 = プリスケール分周比

11111 = 予約済み最小インターバル (1:32)

⋮

10011 = 予約済み最小インターバル (1:32)

10010 = 1:8388608 (2^{23}) (インターバル公称 256 s)

10001 = 1:4194304 (2^{22}) (インターバル公称 128 s)

10000 = 1:2097152 (2^{21}) (インターバル公称 64 s)

01111 = 1:1048576 (2^{20}) (インターバル公称 32 s)

01110 = 1:524288 (2^{19}) (インターバル公称 16 s)

01101 = 1:262144 (2^{18}) (インターバル公称 8 s)

01100 = 1:131072 (2^{17}) (インターバル公称 4 s)

01011 = 1:65536(インターバル公称 2 s) (リセット値)

01010 = 1:32768(インターバル公称 1 s)

01001 = 1:16384(インターバル公称 512 ms)

01000 = 1:8192(インターバル公称 256 ms)

00111 = 1:4096(インターバル公称 128 ms)

00110 = 1:2048(インターバル公称 64 ms)

00101 = 1:1024(インターバル公称 32 ms)

00100 = 1:512(インターバル公称 16 ms)

00011 = 1:256(インターバル公称 8 ms)

00010 = 1:128(インターバル公称 4 ms)

00001 = 1:64(インターバル公称 2 ms)

00000 = 1:32(インターバル公称 1 ms)

bit 0 **SWDTEN**: ウォッチドッグ タイマのソフトウェア有効 / 無効ビット

WDTE<1:0> = 1x の場合：

このビットを無視する

WDTE<1:0> = 01 の場合：

1 = WDT を有効にする

0 = WDT を無効にする

WDTE<1:0> = 00 の場合：

このビットを無視する

Note 1: 時間は近似値です。WDT の時間は 31 kHz LFINTOSC に基づいて決まります。

PIC16(L)F1764/5/8/9

表 9-3: ウォッチドッグ タイマ関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
OSCCON	SPLLEN	IRCF<3:0>			—		SCS<1:0>		86
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
WDTCON	—	—	WDTPS<4:0>					SWDTEN	117

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し。網掛けの部分はウォッチドッグ タイマでは使いません。

表 9-4: ウォッチドッグ タイマ関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ 内容記載 ページ
CONFIG1	13:8	—	—	FCMEN	IESO	$\overline{CLKOUTEN}$	BOREN<1:0>	—	65	
	7:0	\overline{CP}	MCLRE	\overline{PWRTE}	WDTE<1:0>	FOOSC<2:0>				

凡例: — = 未実装、「0」として読み出し。網掛けの部分はウォッチドッグ タイマでは使いません。

10.0 フラッシュ プログラムメモリ制御

フラッシュ プログラムメモリは、通常動作中 VDD のレンジ全体で読み書きが可能です。プログラムメモリは特殊機能レジスタ (SFR) を使って間接的にアドレス指定します。プログラムメモリへのアクセスには以下の SFR を使います。

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

プログラムメモリにアクセスする場合、PMDATH:PMDATL レジスタペアの2バイトワードに読み書きする14ビットのデータを格納し、PMADRH:PMADRL レジスタペアの2バイトワードにプログラムメモリのアクセス先となる15ビットのアドレスを格納します。

書き込み時間は、内蔵タイマで制御します。書き込み/消去電圧は、デバイスの動作電圧レンジよりも高い値で、内蔵チャージポンプで生成されます。

フラッシュ プログラムメモリは、コード保護 (コンフィグレーションワードの \overline{CP} ビット) と書き込み保護 (コンフィグレーションワードの WRT<1:0> ビット) の2つの方法で保護できます。

コード保護 ($\overline{CP} = 0$) ⁽¹⁾ は、外付けのデバイス プログラムによるフラッシュ プログラムメモリに対するアクセス、すなわち読み書きを禁止します。ただし、自己書き込みと消去機能には影響を与えません。コード保護をリセットするには、デバイス プログラムによるデバイスのバルク消去以外に方法がありません。バルク消去では、フラッシュ プログラムメモリ、コンフィグレーション ビット、ユーザ ID の全てがクリアされます。

書き込み保護は WRT<1:0> ビットで設定し、フラッシュ プログラムメモリの一部または全体に対して自己書き込みと消去を禁止します。書き込み保護は、デバイス プログラムによるデバイスの読み書き、消去機能には影響を与えません。

Note 1: フラッシュ プログラムメモリのアレイ全体に対してコード保護を有効にするには、コンフィグレーションワードの \overline{CP} ビットをクリアします。

10.1 PMADRL および PMADRH レジスタ

PMADRH:PMADRL レジスタペアは、プログラムメモリの最大 32K ワードに対するアドレスを指定できます。プログラムメモリのアドレス値を選択する場合、アドレスの MSB が PMADRH レジスタへ書き込まれ、LSB が PMADRL レジスタへ書き込まれます。

10.1.1 PMCON1 および PMCON2 レジスタ

PMCON1 は、フラッシュ プログラムメモリに対するアクセスを制御するレジスタです。

RD 制御ビットと WR 制御ビットで、それぞれ読み出しと書き込みを開始します。これらのビットは、ソフトウェアではセットのみ可能でクリアはできません。読み出しまたは書き込み動作が完了した時点で、ハードウェアでクリアされます。WR ビットをソフトウェアでクリアできないようにする事で、書き込み動作が誤って中断されるのを防いでいます。

WREN ビットをセットすると、書き込み動作が許可されます。WREN ビットは、電源投入時にクリアされます。動作中にリセットが発生して書き込み動作が中断した場合、WRERR ビットがセットされます。このような場合、リセット後に WRERR ビットを確認して適切なエラー処理ルーチンを実行できます。

PMCON2 レジスタは書き込み専用レジスタです。PMCON2 レジスタを読み出すと、常に「0」が返されます。

プログラムメモリへの書き込みを有効にするには、PMCON2 レジスタに特定のパターン (ロック解除シーケンス) を書き込む必要があります。このロック解除シーケンスを必要とする事で、プログラムメモリの書き込みラッチとフラッシュ プログラムメモリに対する偶発的な書き込みを防止しています。

10.2 フラッシュ プログラムメモリの概要

フラッシュ プログラムメモリに対して消去と書き込みを実行する場合、その構造を理解しておく事が大切です。フラッシュ プログラムメモリは行単位で構成されています。1行は一定数の14ビット プログラムメモリワードで構成されています。この1行が、ユーザソフトウェアで消去できる最小サイズです。

ユーザは、一度消去した行の一部または全体に対して再書き込みを実行できます。プログラムメモリの行にデータを書き込むには、14ビット幅のデータ書き込みラッチに対して書き込みを実行します。書き込みラッチにユーザは直接アクセスできませんが、PMDATH:PMDATL レジスタペアへの順次書き込みで、書き込みラッチにデータを書き込む事ができます。

Note: 既に書き込み済みの行の一部のみを書き換えたい場合、行全体の内容を読み出し、RAM に保存してから消去を実行します。その後、新しいデータと変更しないデータを書き込みラッチに書き込み、フラッシュ プログラムメモリの行を再プログラムします。しかし、未プログラムの位置には、最初に消去せずに書き込みを実行できます。この場合、既にプログラム済みの他のメモリ位置の内容を保存して再書き込みする必要はありません。

フラッシュ プログラムメモリの行消去サイズと書き込みラッチ数は、表 10-1 を参照してください。

PIC16(L)F1764/5/8/9

表 10-1: 各デバイスのフラッシュメモリの構成

デバイス	行消去 (ワード)	書き込みラッチ (ワード)
PIC16(L)F1764	32	32
PIC16(L)F1765		
PIC16(L)F1768		
PIC16(L)F1769		

10.2.1 フラッシュ プログラムメモリの読み出し

プログラムメモリからの読み出しは、以下のステップで実行します。

1. 読み出すアドレスを PMADRH:PMADRL レジスタペアに書き込む。
2. PMCON1 レジスタの CFGS ビットをクリアする。
3. PMCON1 レジスタの RD 制御ビットをセットする。

読み出し制御ビットをセットした後、プログラムメモリフラッシュコントローラは2番目の命令サイクルでデータを読み出します。このため、「BSF PMCON1, RD」命令に続く2番目の命令は無視されます。データは、その次のサイクルで PMDATH:PMDATL レジスタペアに格納されます。従って、これ以降の命令で2バイトとして読み出す事ができます。

PMDATH:PMDATL レジスタペアに格納された値は、次の読み出し動作か、ユーザが書き込みを実行するまで保持されます。

Note: プログラムメモリの読み出し後の2つの命令は、NOP とする必要があります。これにより、RD ビットがセットされた直後の命令でユーザが2サイクル命令を実行する事を防ぎます。

図 10-1: フラッシュ プログラムメモリの読み出しフローチャート

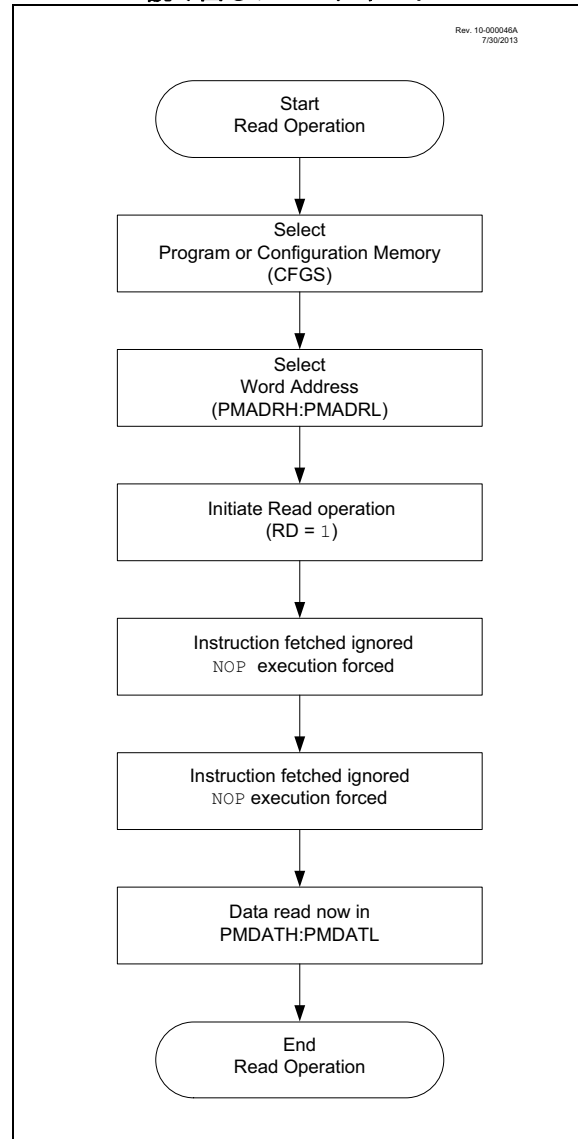
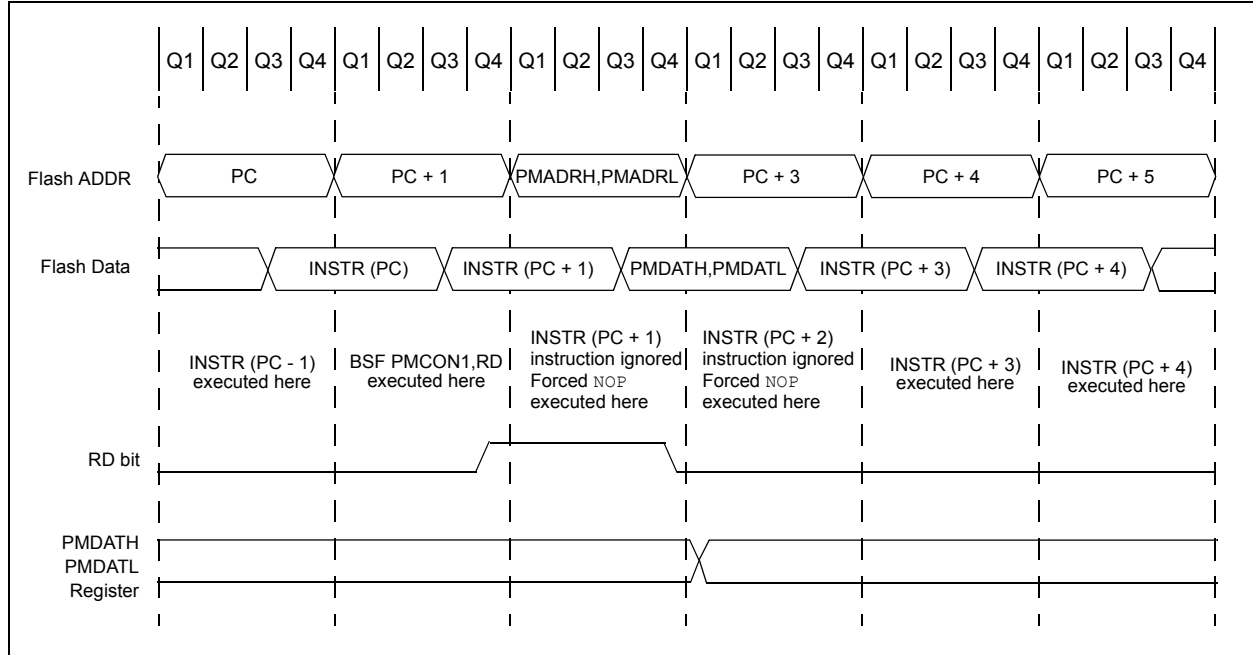


図 10-2: フラッシュ プログラムメモリの読み出しサイクルの実行



例 10-1: フラッシュ プログラムメモリの読み出し

```

* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI :PROG_ADDR_LO
* data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO

  BANKSEL  PMADRL          ; Select Bank for PCON registers
  MOVLW   PROG_ADDR_LO    ;
  MOVWF   PMADRL          ; Store LSB of address
  MOVLW   PROG_ADDR_HI    ;
  MOVWF   PMADRH         ; Store MSB of address

  BCF     PMCON1,CFGSS    ; Do not select Configuration Space
  BSF     PMCON1,RD       ; Initiate read
  NOP     ; Ignored ( 図 10-1)
  NOP     ; Ignored ( 図 10-1)

  MOVF    PMDATL,W        ; Get LSB of word
  MOVWF   PROG_DATA_LO   ; Store in user location
  MOVF    PMDATH,W        ; Get MSB of word
  MOVWF   PROG_DATA_HI   ; Store in user location
    
```

PIC16(L)F1764/5/8/9

10.2.2 フラッシュメモリのロック解除シーケンス

ロック解除シーケンスは、意図しない自己書き込みまたは消去からフラッシュ プログラムメモリを保護する仕組みです。以下の全ての動作を正常に完了するには、このシーケンスを中断する事なく実行し、完了させる必要があります。

- 行消去
- プログラムメモリ書き込みラッチへの書き込み
- プログラムメモリ書き込みラッチからプログラムメモリへの書き込み
- プログラムメモリ書き込みラッチからユーザIDへの書き込み

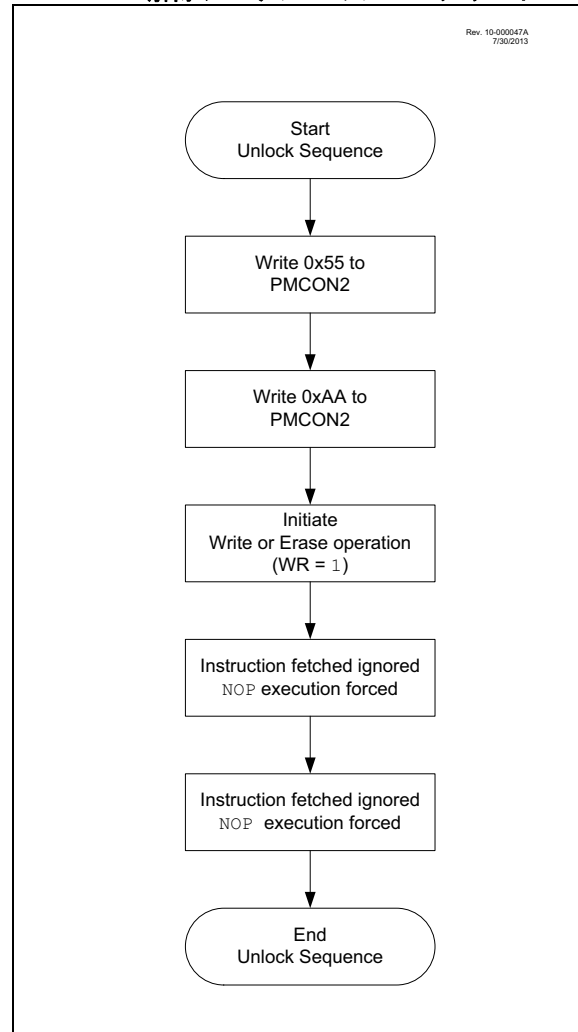
ロック解除シーケンスは、以下のステップで構成されます。

1. PMCON2 への 55h 書き込み
2. PMCON2 への AAh 書き込み
3. PMCON1 の WR ビットのセット
4. NOP 命令
5. NOP 命令

WR ビットがセットされると、プロセッサは常に 2 つの NOP 命令を実行します。行消去または行書き込み動作中、プロセッサは内部動作をストールさせ (2 ms typ.)、動作が完了してから次の命令を再開します。動作がプログラムメモリ書き込みラッチへの書き込みの場合、プロセッサは常に 2 つの NOP 命令を実行してから次の命令を継続します。

ロック解除シーケンスが中断される事のないように、シーケンスの実行前にグローバル割り込みを無効にして、完了後に再度有効にします。

図 10-3: フラッシュ プログラムメモリ ロック解除シーケンスのフローチャート



10.2.3 フラッシュ プログラムメモリの消去

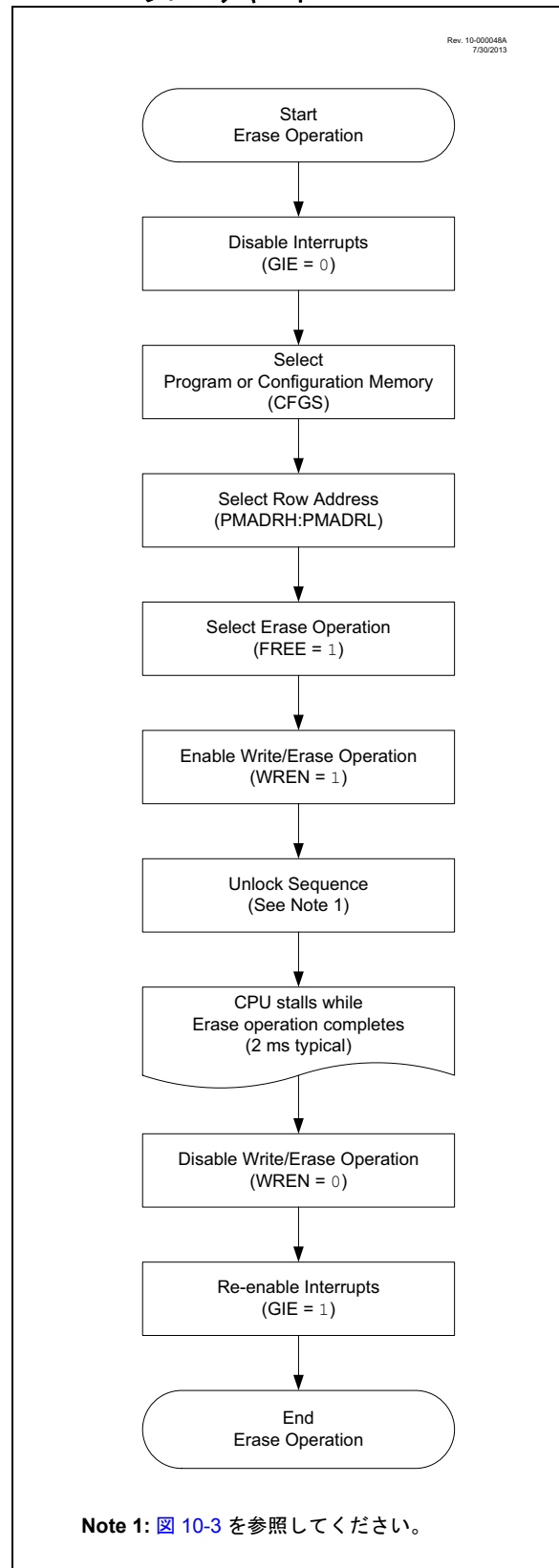
コード実行時、プログラムメモリの消去は行単位でのみ可能です。行を消去するには、以下のステップを実行します。

1. 消去する行内のいずれかのアドレスを PMADRH:PMADRL レジスタペアに書き込む。
2. PMCON1 レジスタの CFGS ビットをクリアする。
3. PMCON1 レジスタの FREE および WREN ビットをセットする。
4. PMCON2 レジスタに 55h、AAh を順に書き込む (フラッシュプログラミングロック解除シーケンス)。
5. PMCON1 レジスタの WR 制御ビットをセットして消去を開始する。

例 10-2 を参照してください。

「BSF PMCON1,WR」命令の後、プロセッサが消去動作を完了するまでに 2 サイクル必要です。このため、WR ビットをセットした直後は、2 つの NOP 命令を実行する必要があります。プロセッサは 2 ms (typ.) の消去時間の間、内部動作を停止します。クロックと周辺モジュールは動作を継続するため、これはスリープとは異なります。消去サイクル後、プロセッサは PMCON1 書き込み命令後の 3 番目の命令から動作を再開します。

図 10-4: フラッシュ プログラムメモリ消去のフローチャート



PIC16(L)F1764/5/8/9

例 10-2: プログラムメモリの 1 行消去

```
; This row erase routine assumes the following:
; 1.A valid address within the erase row is loaded in ADDRH:ADDRL
; 2.ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)

        BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL PMADRL
        MOVF     ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF   PMADRL
        MOVF     ADDRH,W        ; Load upper 6 bits of erase address boundary
        MOVWF   PMADRH
        BCF     PMCON1,CFG5      ; Not configuration space
        BSF     PMCON1,FREE      ; Specify an erase operation
        BSF     PMCON1,WREN      ; Enable writes

        MOVLW   55h             ; Start of required sequence to initiate erase
        MOVWF   PMCON2          ; Write 55h
        MOVLW   0AAh           ;
        MOVWF   PMCON2          ; Write AAh
        BSF     PMCON1,WR       ; Set WR bit to begin erase
        NOP                    ; NOP instructions are forced as processor starts
        NOP                    ; row erase of program memory.
        ;
        ; The processor stalls until the erase process is complete
        ; after erase processor continues with 3rd instruction

        BCF     PMCON1,WREN     ; Disable writes
        BSF     INTCON,GIE      ; Enable interrupts
```

Required
Sequence

10.2.4 フラッシュ プログラムメモリへの書き込み

プログラムメモリに書き込むには、以下のステップを実行します。

1. 書き込む行のアドレスをPMADRH:PMADRLに書き込む。
2. 各書き込みラッチにデータを書き込む。
3. フラッシュ プログラムメモリへの書き込み動作を開始する。
4. 全てのデータの書き込みが完了するまでステップ 1～3 を繰り返す。

プログラムメモリに書き込みを実行する際は、書き込み先に何もデータが書き込まれていないか、書き込まれている場合、事前に消去しておく必要があります。プログラムメモリの消去は行単位でのみ可能です。書き込み開始時に自動的に消去が実行される事はありません。

プログラムメモリへの書き込みは、1 ワードずつまたは複数ワードを一度に書き込む事ができます。一度に書き込む事ができる最大ワード数は、書き込みラッチの数と同じです。詳細は、[図 10-5](#) (32 個の書き込みラッチによるプログラムメモリへの行書き込み) を参照してください。

PMADRH:PMADRL レジスタペアは、上位 10 ビット (PMADRH<6:0>:PMADRL<7:5>) で書き込み先のフラッシュ行アドレス境界を定義し、下位 5 ビット (PMADRL<4:0>) でどの書き込みラッチにデータを書き込むかを定義します。この境界を越えて書き込む事はできません。プログラムメモリへの書き込みが完了したら、書き込みラッチのデータは 0x3FFF にリセットされます。

書き込みラッチにデータを書き込んでプログラムメモリの行書き込みを実行するには、以下のステップを実行する必要があります。このステップは大きく 2 つの部分に分かれます。まず、LWLO ビットをセットし (= 1)、ロック解除シーケンスを実行し、PMDATH:PMDATL のデータを各書き込みラッチに書き込んでいきます。最後の書き込みラッチにワードを書き込む際は、LWLO ビットをクリア (= 0) してロック解除シーケンスを実行します。これによってプログラミング動作が始まり、全てのラッチの内容がフラッシュ プログラムメモリに書き込まれます。

Note: 書き込みラッチへのデータ書き込み、またはフラッシュ書き込みの動作を始めるには、特定のロック解除シーケンスが必要です。ロック解除シーケンスが中断された場合、ラッチまたはプログラムメモリへの書き込みは始まりません。

1. PMCON1レジスタのWRENビットをセットする。
2. PMCON1レジスタのCFG5ビットをクリアする。
3. PMCON1レジスタのLWLOビットをセットする。PMCON1レジスタのLWLOビットが「1」の場合、書き込みシーケンスを実行しても書き込みラッチにデータが書き込まれるだけでフラッシュプログラムメモリへの書き込みは始まりません。
4. 書き込み先のアドレスをPMADRH:PMADRLレジスタペアに書き込む。
5. プログラムメモリに書き込むデータをPMDATH:PMDATLレジスタペアに書き込む。
6. ロック解除シーケンスを実行する ([セクション 10.2.2 「フラッシュメモリのロック解除シーケンス」](#))。以上で書き込みラッチへの書き込みが完了します。
7. 次のアドレスを指すようにPMADRH:PMADRLレジスタペアをインクリメントする。
8. 最後の書き込みラッチを除く全てのラッチに書き込みが完了するまでステップ 5～7 を繰り返す。
9. PMCON1レジスタのLWLOビットをクリアする。PMCON1レジスタのLWLOビットが「0」の場合、書き込みシーケンスを実行するとフラッシュプログラムメモリへの書き込みが始まります。
10. プログラムメモリに書き込むデータをPMDATH:PMDATLレジスタペアに書き込む。
11. ロック解除シーケンスを実行する ([セクション 10.2.2 「フラッシュメモリのロック解除シーケンス」](#))。以上でプログラムメモリラッチの内容が全てフラッシュ プログラムメモリに書き込まれました。

Note: プログラムメモリ書き込みラッチは、書き込みまたは消去動作が完了するたびにブランク状態(0x3FFF)にリセットされます。このため、プログラムメモリ書き込みラッチの全てにデータを書き込む必要はありません。データを書き込まなければ、ラッチはブランク状態のままです。

例 10-3 に、書き込みシーケンス全体の例を示します。PMADRH:PMADRL レジスタペアには、最初のアドレスが書き込まれます。データは間接アドレス指定で書き込まれます。

図 10-5: 32個の書き込みラッチによるフラッシュプログラムメモリへのブロック書き込み

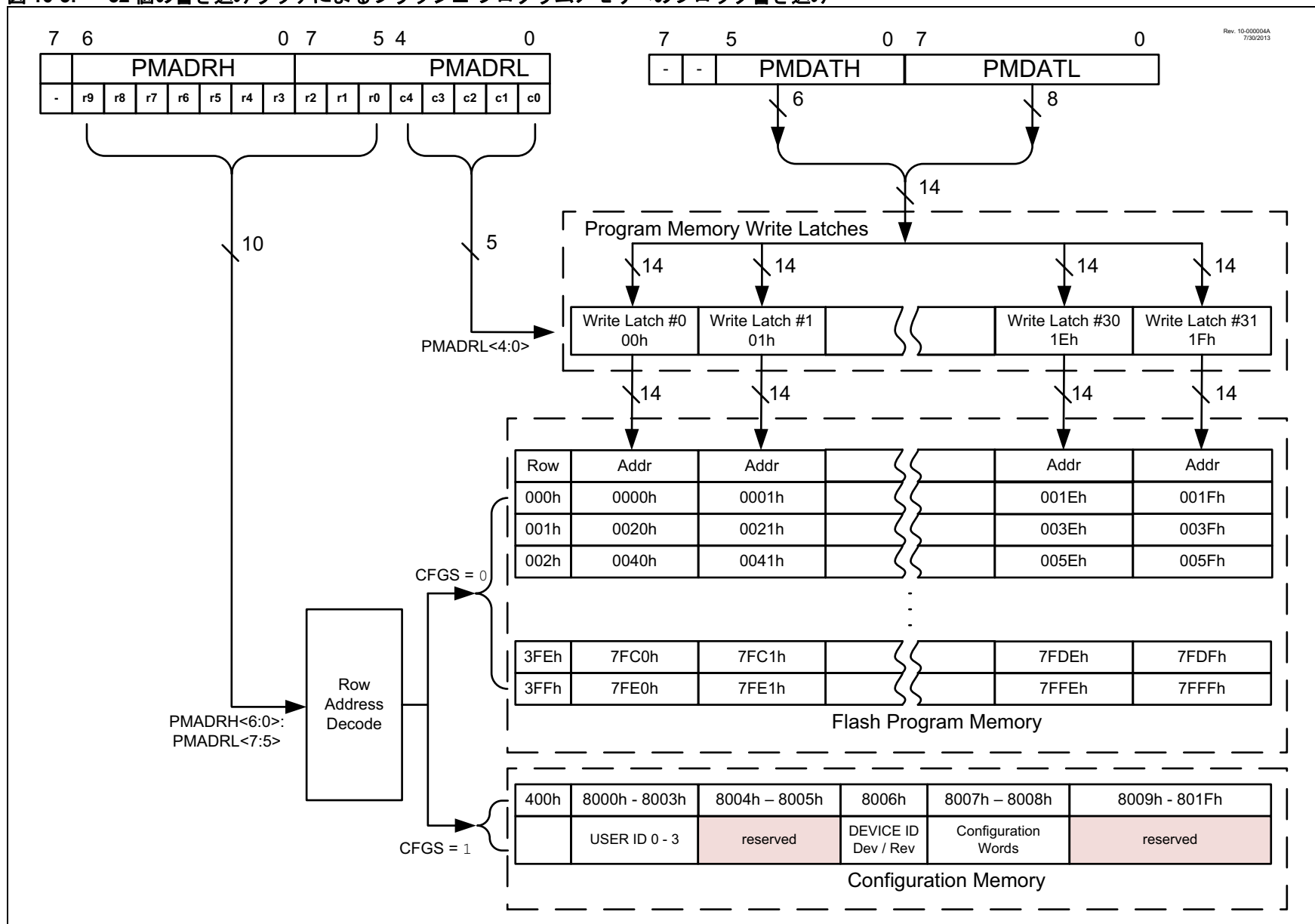
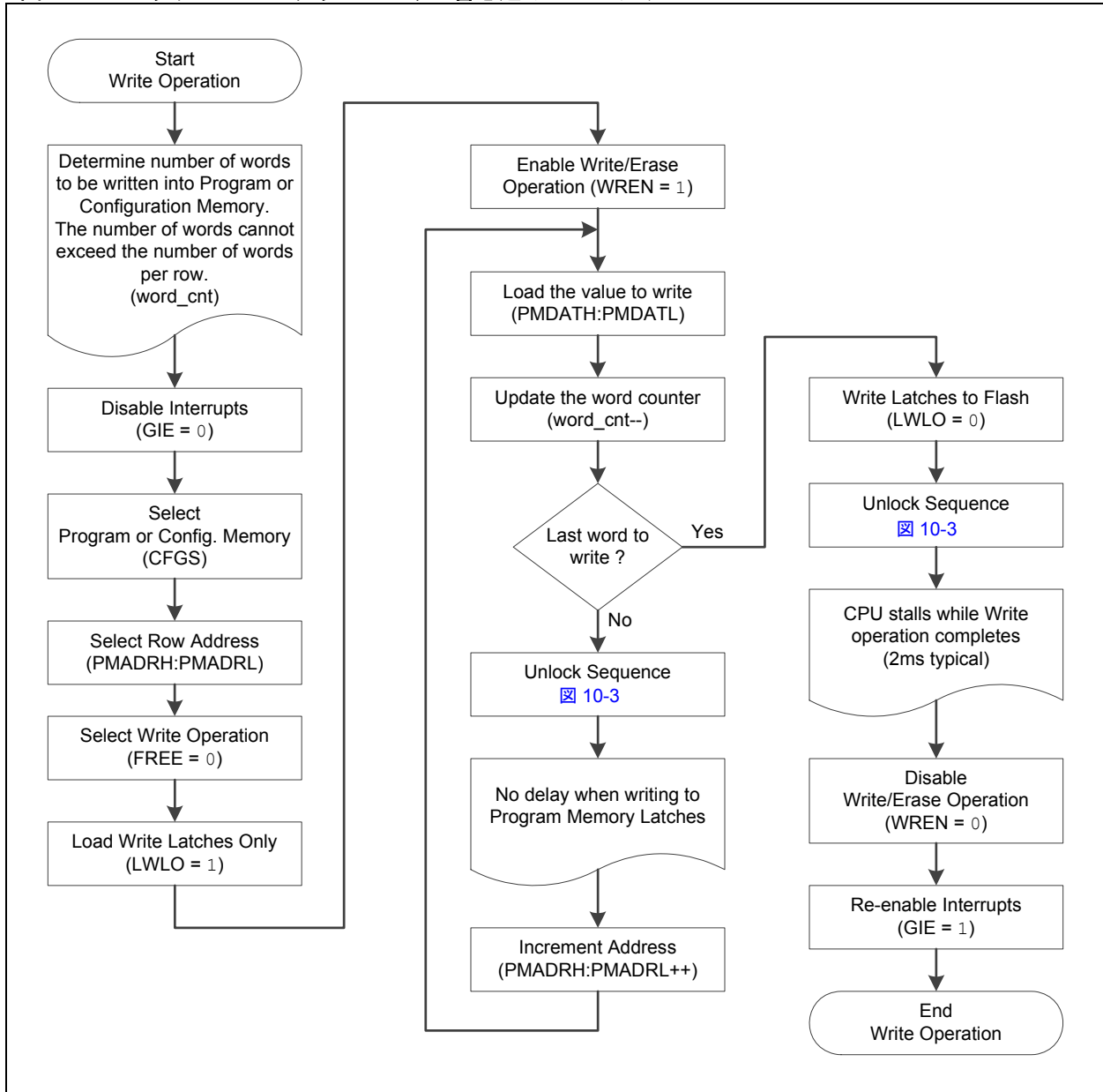


図 10-6: フラッシュ プログラムメモリの書き込みフローチャート



PIC16(L)F1764/5/8/9

例 10-3: フラッシュ プログラムメモリへの書き込み

```
; This write routine assumes the following:
; 1.64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2.Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3.A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4.ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
    BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
    BANKSEL PMADRH          ; Bank 3
    MOVF    ADDRH,W         ; Load initial address
    MOVWF   PMADRH          ;
    MOVF    ADDRHL,W        ;
    MOVWF   PMADRL          ;
    MOVLW   LOW DATA_ADDR  ; Load initial data address
    MOVWF   FSR0L           ;
    MOVLW   HIGH DATA_ADDR ; Load initial data address
    MOVWF   FSR0H           ;
    BCF     PMCON1,CFGSS    ; Not configuration space
    BSF     PMCON1,WREN     ; Enable writes
    BSF     PMCON1,LWLO     ; Only Load Write Latches

LOOP
    MOVIW   FSR0++          ; Load first data byte into lower
    MOVWF   PMDATH          ;
    MOVIW   FSR0++          ; Load second data byte into upper
    MOVWF   PMDATH          ;

    MOVF    PMADRL,W        ; Check if lower bits of address are '00000'
    XORLW   0x1F            ; Check if we're on the last of 32 addresses
    ANDLW   0x1F            ;
    BTFSC   STATUS,Z        ; Exit if last of 32 words,
    GOTO    START_WRITE     ;

    [ Required Sequence
    MOVLW   55h              ; Start of required write sequence:
    MOVWF   PMCON2           ; Write 55h
    MOVLW   0AAh             ;
    MOVWF   PMCON2           ; Write AAh
    BSF     PMCON1,WR        ; Set WR bit to begin write
    NOP     ; NOP instructions are forced as processor
    ; loads program memory write latches
    NOP     ;

    INCF    PMADRL,F         ; Still loading latches Increment address
    GOTO    LOOP            ; Write next latches

START_WRITE
    BCF     PMCON1,LWLO     ; No more loading latches - Actually start Flash program
    ; memory write

    [ Required Sequence
    MOVLW   55h              ; Start of required write sequence:
    MOVWF   PMCON2           ; Write 55h
    MOVLW   0AAh             ;
    MOVWF   PMCON2           ; Write AAh
    BSF     PMCON1,WR        ; Set WR bit to begin write
    NOP     ; NOP instructions are forced as processor writes
    ; all the program memory write latches simultaneously
    NOP     ; to program memory.
    ; After NOPs, the processor
    ; stalls until the self-write process is complete
    ; after write processor continues with 3rd instruction

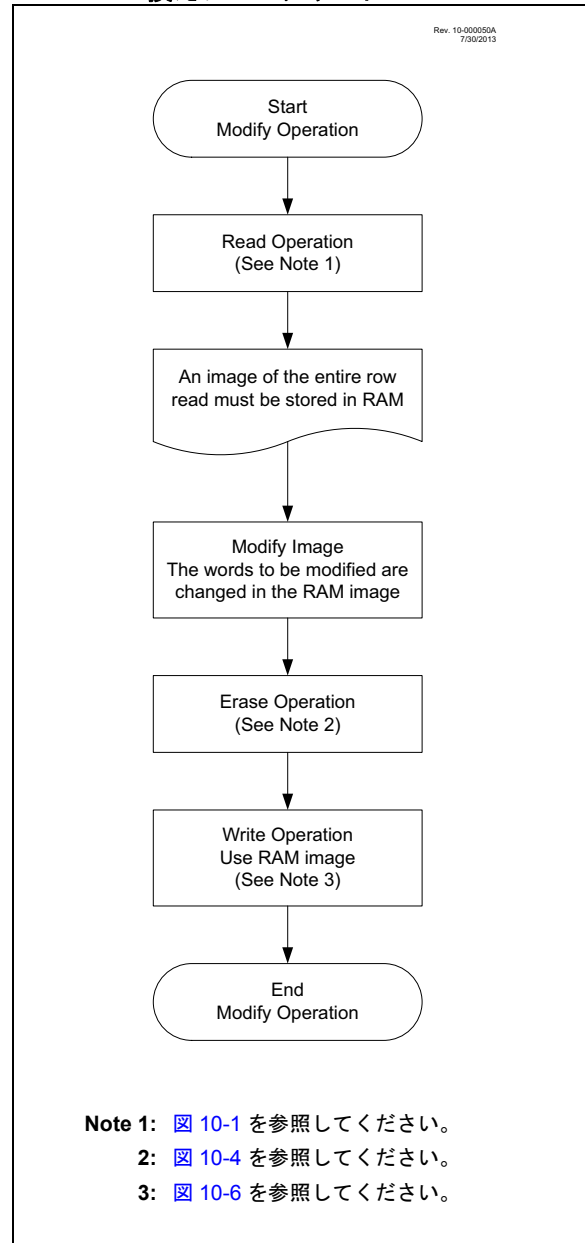
    BCF     PMCON1,WREN     ; Disable writes
    BSF     INTCON,GIE      ; Enable interrupts
```


10.3 フラッシュ プログラムメモリの書き換え

プログラムメモリの行に格納された既存データを保持しながら一部のみを書き換える場合、最初にその行を読み出して RAM イメージに保存しておく必要があります。プログラムメモリを書き換えるには、以下のステップを実行します。

1. 書き換える行の開始アドレスを書き込む。
2. その行のデータを読み出して RAM イメージに保存する。
3. RAM イメージを、新たに書き込むデータが含まれるように更新する。
4. 書き換える行の開始アドレスを書き込む。
5. プログラムメモリの行を消去する。
6. 書き込みラッチに RAM イメージのデータを書き込む。
7. フラッシュ プログラムメモリへの書き込み動作を開始する。

図 10-7: フラッシュ プログラムメモリの書き換えフローチャート



PIC16(L)F1764/5/8/9

10.4 ユーザ ID、デバイス ID、コンフィグレーションワードへのアクセス

PMCON1 レジスタの CFGS = 1 とすると、プログラムメモリの代わりに、ユーザ ID、デバイス ID/ リビジョン ID、コンフィグレーションワードにアクセスできます。これは、PC<15> = 1 で示される領域ですが、全てのアドレスがアクセス可能という訳ではありません。実行可能なアクセスの種類 (読み出し / 書き込み) は、アドレスによって異なります。表 10-2 を参照してください。

表 10-2 に記載したパラメータの範囲外のアドレスに対して読み出しアクセスを開始した場合、PMDATH:PMDATL レジスタペアはクリアされ、読み出し値としては「0」が返されます。

表 10-2: ユーザ ID、デバイス ID、コンフィグレーションワードへのアクセス (CFGS = 1)

アドレス	機能	読み出しアクセス	書き込みアクセス
8000h-8003h	ユーザ ID	可	可
8005h-8006h	デバイス ID/ リビジョン ID	可	不可
8007h-8008h	コンフィグレーションワード 1 および 2	可	不可

例 10-4: コンフィグレーションワードとデバイス ID へのアクセス

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

BANKSEL  PMADRL           ; Select correct Bank
MOVLW    PROG_ADDR_LO    ;
MOVWF    PMADRL          ; Store LSB of address
CLRF     PMADRH          ; Clear MSB of address

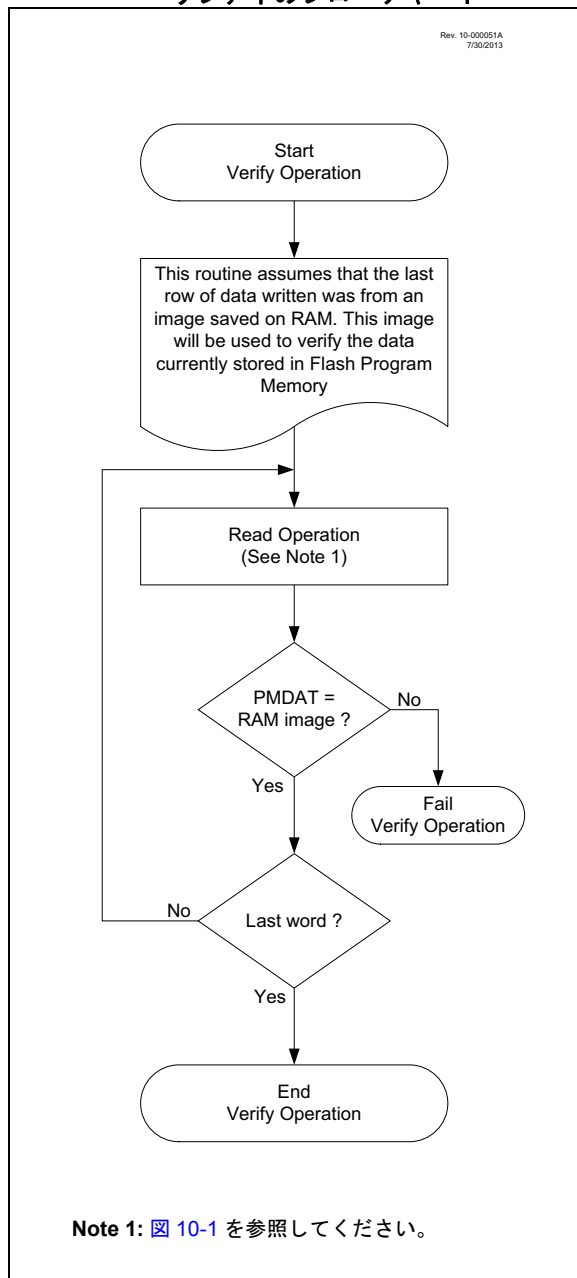
BSF      PMCON1,CFGFS    ; Select Configuration Space
BCF      INTCON,GIE      ; Disable interrupts
BSF      PMCON1,RD       ; Initiate read
NOP      ; Executed (See 10-2)
NOP      ; Ignored (See 10-2)
BSF      INTCON,GIE      ; Restore interrupts

MOVF     PMDATL,W        ; Get LSB of word
MOVWF    PROG_DATA_LO    ; Store in user location
MOVF     PMDATH,W        ; Get MSB of word
MOVWF    PROG_DATA_HI    ; Store in user location
```

10.5 書き込みベリファイ

プログラムメモリへの書き込み後に、正しい値が書き込まれた事を確認（ベリファイ）する事を推奨します。プログラムメモリはページ全体として格納されるため、最後の書き込みの完了後に、プログラムメモリの内容とRAM内の目的とするデータを比較します。

図 10-8: フラッシュ プログラムメモリベリファイのフローチャート



PIC16(L)F1764/5/8/9

10.6 レジスタ定義：フラッシュ プログラムメモリ制御

レジスタ 10-1: PMDATL: プログラムメモリ データ下位バイトレジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **PMDAT<7:0>**: プログラムメモリの下位ビットの読み書き値

レジスタ 10-2: PMDATH: プログラムメモリ データ上位バイトレジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—		PMDAT<13:8>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装**: 「0」として読み出し

bit 5-0 **PMDAT<13:8>**: プログラムメモリの上位ビットの読み書き値

レジスタ 10-3: PMADRL: プログラムメモリ アドレス下位バイトレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PMADR<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **PMADR<7:0>**: プログラムメモリ アドレスの下位ビットを指定する

レジスタ 10-4: PMADRH: プログラムメモリ アドレス上位バイトレジスタ

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—(1)	PMADR<14:8>						
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7 **未実装:** 「1」として読み出し

bit 6-0 **PMADR<14:8>:** プログラムメモリ アドレスの上位ビットを指定する

Note 1: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

レジスタ 10-5: PMCON1: プログラムメモリ制御レジスタ 1

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q ⁽²⁾	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
— ⁽¹⁾	CFGS	LWLO ⁽³⁾	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
S = ビットはセットのみ可能	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」 = ビットはセット	「0」 = ビットはクリア	HC = ビットはハードウェアでクリア

- bit 7 **未実装:** 「1」として読み出し
- bit 6 **CFGS:** コンフィグレーション選択ビット
 1 = コンフィグレーション、ユーザ ID、デバイス ID レジスタにアクセスする
 0 = フラッシュ プログラムメモリにアクセスする
- bit 5 **LWLO:** 書き込みラッチのみ書き込みビット⁽³⁾
 1 = 次のWRコマンドで、アドレス指定されたプログラムメモリ書き込みラッチに対してのみ書き込み/更新を実行する
 0 = 次のWRコマンドで、アドレス指定されたプログラムメモリ書き込みラッチに対する書き込み/更新を実行した後、全てのプログラムメモリ書き込みラッチの内容をプログラムメモリに書き込む
- bit 4 **FREE:** プログラム フラッシュ消去イネーブルビット
 1 = 次の WR コマンドで消去動作を実行する (完了後、ハードウェアでクリアされる)
 0 = 次の WR コマンドで書き込みを実行する
- bit 3 **WRERR:** 書き込み / 消去エラーフラグ ビット
 1 = 書き込み / 消去シーケンスが正しく実行されなかった、または途中で終了した (このビットは、WR ビットを「1」に設定する命令を発行すると自動的にセットされる)
 0 = 書き込みまたは消去動作は正常に完了した
- bit 2 **WREN:** 書き込み / 消去イネーブルビット
 1 = 書き込み / 消去サイクルを許可する
 0 = プログラム フラッシュの書き込み / 消去を禁止する
- bit 1 **WR:** 書き込み制御ビット
 1 = プログラム フラッシュの書き込み / 消去動作を開始する。
 動作は自己タイマで実行され、ビットは動作完了時にハードウェアでクリアされる。
 WR ビットはソフトウェアではセットのみ可能で、クリアできない
 0 = フラッシュの書き込み / 消去動作が完了し、非アクティブ状態である
- bit 0 **RD:** 読み出し制御ビット
 1 = プログラム フラッシュの読み出しを開始する。読み出し動作は1サイクルで完了する。RDビットはハードウェアでクリアされる。ソフトウェアではセットのみ可能で、クリアできない
 0 = プログラム フラッシュの読み出しを開始しない

- Note** 1: 未実装、「1」として読み出します。
 2: プログラムメモリの書き込みまたは消去動作が開始すると (WR = 1)、WRERR ビットはハードウェアで自動的にセットされます。
 3: プログラムメモリの消去動作中 (FREE = 1)、LWLO ビットは無視されます。

PIC16(L)F1764/5/8/9

レジスタ 10-6: PMCON2: プログラムメモリ制御レジスタ 2

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
プログラムメモリ制御レジスタ 2							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 S = ビットはセットのみ可能 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
 「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 フラッシュメモリ ロック解除パターンビット

書き込みロックを解除するには、このビットに 55h、AAh を順に書き込んだ後、PMCON1 レジスタの WR ビットをセットする必要があります。このレジスタへ書き込まれた値は、書き込みロックの解除に使われます。これらの書き込み動作には一定のタイミング要件があります。

表 10-3: フラッシュ プログラムメモリ関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ内容記載ページ
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PMCON1	— ⁽¹⁾	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	134
PMCON2	プログラムメモリ制御レジスタ 2								135
PMADRL	PMADRL<7:0>								132
PMADRH	— ⁽¹⁾	PMADRH<6:0>							133
PMDATL	PMDATL<7:0>								132
PMDATH	—	—	PMDATH<5:0>					132	

凡例: — = 未実装、「0」として読み出します。網掛けの部分はフラッシュ プログラムメモリでは使いません。

Note 1: 未実装、「1」として読み出します。

表 10-4: フラッシュ プログラムメモリ関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ内容記載ページ
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	67
	7:0	ZCD	—	—	—	—	PPS1WAY	WRT<1:0>		

凡例: — = 未実装、「0」として読み出します。網掛けの部分はフラッシュ プログラムメモリでは使いません。

PIC16(L)F1764/5/8/9

11.0 I/O ポート

各ポートは、6つの標準レジスタを使って動作します。これらのレジスタは以下の通りです。

- TRISx レジスタ (データ方向)
- PORTx レジスタ (デバイスピンのレベルの読み取り)
- LATx レジスタ (出力ラッチ)
- INLVLx (入力レベル制御)
- ODCONx レジスタ (オープンドレイン)
- SLRCONx レジスタ (スルーレート)

ポートによっては、追加レジスタが使えます。これらのレジスタは以下の通りです。

- ANSELx (アナログ選択)
- WPUx (弱プルアップ)

通常、あるポートピンで周辺モジュールを有効にすると、そのピンは汎用出力として使う事はできません。しかし、ピンの読み出しは可能です。

表 11-1: 各デバイスで利用可能なポート

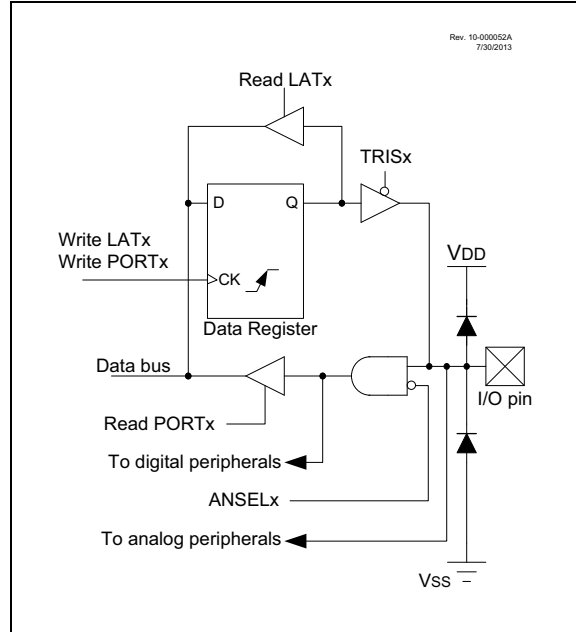
デバイス	PORTA	PORTB	PORTC
PIC16(L)F1764	•		•
PIC16(L)F1765	•		•
PIC16(L)F1768	•	•	•
PIC16(L)F1769	•	•	•

データラッチ (LATx レジスタ) は、I/O ピンが駆動している値の Read-Modify-Write 動作に便利です。

LATx レジスタへの書き込み動作は、対応する PORTx レジスタへの書き込みと同じ効果があります。LATx レジスタを読み出すと、I/O ポートラッチに格納されている値が読み出されるのに対し、PORTx レジスタを読み出した場合、実際の I/O ピンの値が読み出されます。

アナログ入力をサポートしているポートには、対応する ANSELx レジスタがあります。ANSEL ビットをセットすると、そのビットに対応するデジタル入力バッファが無効になります。入力バッファが無効になると、そのピンに論理 High と Low の中間のアナログ信号レベルが入力されても論理入力回路に過大電流が流れるのを防ぐ事ができます。図 11-1 に、他の周辺モジュールへのインターフェイスを省略した、汎用 I/O ポートの簡易モデルを示します。

図 11-1: 汎用 I/O ポートの動作



11.1 PORTA レジスタ

11.1.1 データレジスタ

PORTA は 6 ビット幅の双方向ポートです。これに対応するデータ方向レジスタが、TRISA([レジスタ 11-2](#)) です。TRISA ビットをセットする (= 1) と、対応する PORTA ピンが入力になります (すなわち、出力ドライバが無効になります)。TRISA ビットをクリアする (= 0) と、対応する PORTA のピンが出力になります (すなわち、出力ドライバが有効になり、出力ラッチの内容が選択したピンに出力されます)。例外として、RA3 は入力専用であり、その TRIS ビットは常に「1」として読み出されます。[例 11-1](#) に PORTA の初期化方法を示します。

PORTA レジスタ ([レジスタ 11-1](#)) を読み出すとピンのステータスが読み出され、PORTA レジスタに書き込むと PORT ラッチに書き込まれます。書き込み動作は全て Read-Modify-Write です。従って、ポートへの書き込み時にはまずポートピンが読み出され、この値が変更されてから PORT データラッチ (LATA) に書き込まれます。

11.1.2 方向制御

ピンをアナログ入力として使う場合も、TRISA レジスタ ([レジスタ 11-2](#)) が PORTA ピンの出力ドライバを制御します。これらのピンをアナログ入力として使う場合、必ず TRISA レジスタのビットをセットしたままにします。アナログ入力として設定された I/O ピンは常に「0」として読み出されます。

11.1.3 オープンドレイン制御

ODCONA レジスタ ([レジスタ 11-6](#)) は、ポートのオープンドレイン機能を制御します。オープンドレインの動作は、各ピンで個別に選択します。ODCONA ビットをセットすると、対応するポート出力は電流のシンクのみ可能なオープンドレイン ドライバとなります。ODCONA ビットをクリアすると、対応するポート出力ピンは電流のソースとシンクが可能な標準のプッシュプル ドライバとなります。

11.1.4 スルーレート制御

SLRCONA レジスタ ([レジスタ 11-7](#)) は、各ポートピンのスルーレートを制御します。スルーレート制御は、各ポートピンで個別に選択できます。SLRCONA ビットをセットした場合、対応するポートピンは制限されたスルーレートで駆動されます。SLRCONA ビットをクリアした場合、対応するポートピンは最大スルーレートで駆動されます。

11.1.5 入力しきい値制御

INLVLA レジスタ ([レジスタ 11-8](#)) は、各 PORTA 入力ピンの入力電圧しきい値を制御します。シュミットトリガ CMOS か TTL 互換しきい値を選択できます。入力しきい値は、PORTA レジスタの読み出し値、そして状態変化割り込み有効時には割り込みを生成するレベルを決定する重要な要素です。しきい値レベルの詳細は、[表 36-4: I/O ポート](#)を参照してください。

Note: 入力しきい値の選択は、全ての周辺モジュールを無効にした状態で変更します。アクティブなモジュールがある時にしきい値レベルを変更すると、入力ピンでの実際の電圧レベルに関係なく、そのピンに関連する状態遷移が誤って生成される場合があります。

11.1.6 アナログ制御

ANSELA レジスタ ([レジスタ 11-4](#)) は、I/O ピンの入力モードをアナログに設定する場合に使います。ANSELA ビットを High に設定したピンに対してデジタル読み出しを実行すると、全て「0」として読み出され、ピンのアナログ機能が正しく動作します。

ANSELA ビットの状態は、デジタル出力機能には影響を与えません。TRIS をクリアして ANSEL をセットしたピンはデジタル出力として動作しますが、入力モードはアナログです。このように設定されたポートに対して Read-Modify-Write 命令を実行すると予期しない挙動を引き起こす事があります。

Note: ANSELA ビットのリセット後の既定値はアナログモードです。任意のピンをデジタル汎用入力または周辺入力として使うには、対応する ANSEL ビットをユーザーソフトウェアで「0」に初期化する必要があります。

例 11-1: PORTA の初期化

```

; This code example illustrates
; initializing the PORTA register.The
; other ports are initialized in the same
; manner.

BANKSEL PORTA      ;
CLRF PORTA         ;Init PORTA
BANKSEL LATA       ;Data Latch
CLRF LATA          ;
BANKSEL ANSELA     ;
CLRF ANSELA        ;digital I/O
BANKSEL TRISA      ;
MOVLW B'00111000' ;Set RA<5:3> as inputs
MOVWF TRISA        ;and set RA<2:0> as
                  ;outputs
    
```

PIC16(L)F1764/5/8/9

11.1.7 PORTA の機能と出力の優先度

PORTA の各ピンは、他の機能と多重化されています。各ピンはリセット後、既定値である PORT ラッチデータに戻ります。その他の機能はペリフェラルピンセレクトロジックで選択します。詳細は[セクション 12.0 「ペリフェラルピンセレクト \(PPS\) モジュール」](#)を参照してください。

ADC およびコンパレータ入力等のアナログ入力機能はペリフェラルピンセレクトのリストには記載していません。これらの入力は、ANSELA レジスタで I/O ピンをアナログモードに設定すると有効になります。アナログモードの場合、デジタル出力機能は継続してピンを制御できます。

11.2 レジスタ定義 :PORTA

レジスタ 11-1: PORTA: PORTA レジスタ

U-0	U-0	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **RA<5:0>:** PORTA I/O 値のビット⁽¹⁾
 1 = PORT ピン ≥ V_{IH} である
 0 = PORT ピン ≤ V_{IL} である

Note 1: PORTA への書き込みは、実際には対応する LATA レジスタへの書き込み動作です。PORTA レジスタからの読み出しは、実際には I/O ピン値の読み出し動作です。

レジスタ 11-2: TRISA: PORTA 3 ステートレジスタ

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-4 **TRISA<5:4>:** PORTA 3 ステート制御ビット
 1 = PORTA ピンを入力として設定する (3 ステート)
 0 = PORTA ピンを出力として設定する
bit 3 **未実装:** 「1」として読み出し
bit 2-0 **TRISA<2:0>:** PORTA 3 ステート制御ビット
 1 = PORTA ピンを入力として設定する (3 ステート)
 0 = PORTA ピンを出力として設定する

Note 1: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

レジスタ 11-3: LATA: PORTA データラッチ レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-6 **未実装:** 「0」として読み出し
- bit 5-4 **LATA<5:4>:** RA<5:4> 出力ラッチ値のビット (1)
- bit 3 **未実装:** 「0」として読み出し
- bit 2-0 **LATA<2:0>:** RA<2:0> 出力ラッチ値のビット (1)

Note 1: PORTA への書き込みは、実際には対応する LATA レジスタへの書き込み動作です。PORTA レジスタからの読み出しは、実際には I/O ピン値の読み出し動作です。

レジスタ 11-4: ANSELA: PORTA アナログ選択レジスタ

U-0	U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-5 **未実装:** 「0」として読み出し
- bit 4 **ANSA4:** RA4 ピンの機能をアナログまたはデジタルのどちらかに設定します。
1 = アナログ入力: ピンをアナログ入力として割り当て (1)、デジタル入力バッファを無効にする
0 = デジタル I/O: ピンをポートまたはデジタル特殊機能に割り当てる
- bit 3 **未実装:** 「0」として読み出し
- bit 2-0 **ANSA<2:0>:** RA<2:0> の各ピンの機能をアナログまたはデジタルのどちらかに設定します。
1 = アナログ入力: ピンをアナログ入力として割り当て (1)、デジタル入力バッファを無効にする
0 = デジタル I/O: ピンをポートまたはデジタル特殊機能に割り当てる

Note 1: アナログ入力として設定する場合、ピンの電圧を外部から制御するには、対応する TRIS ビットをセットする必要があります。

レジスタ 11-5: WPUA: PORTA 弱プルアップ レジスタ

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **WPUA<5:0>:** 弱プルアップ レジスタビット (1), (2)
1 = プルアップを有効にする
0 = プルアップを無効にする

- Note 1:** 個別にプルアップを有効にする場合、OPTION_REG レジスタのグローバル $\overline{\text{WPUEN}}$ ビットをクリアする必要があります。
- 2:** ピンが出力として設定されている場合、弱プルアップは自動的に無効になります。

レジスタ 11-6: ODCONA: PORTA オープンドレイン制御レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し

bit 5-4 **ODA<5:4>:** PORTA オープンドレイン イネーブルビット
RA<5:4> の各ピンに対して:
1 = ポートピンの動作をオープンドレイン駆動 (シンクのみ) とする
0 = ポートピンの動作を標準のプッシュプル駆動 (ソースとシンク) とする

bit 3 **未実装:** 「0」として読み出し

bit 2-0 **ODA<2:0>:** PORTA オープンドレイン イネーブルビット
RA<2:0> の各ピンに対して:
1 = ポートピンの動作をオープンドレイン駆動 (シンクのみ) とする
0 = ポートピンの動作を標準のプッシュプル駆動 (ソースとシンク) とする

PIC16(L)F1764/5/8/9

レジスタ 11-7: SLRCONA: PORTA スルーレート制御レジスタ

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-6 **未実装:** 「0」として読み出し
- bit 5-4 **SLRA<5:4>:** PORTA スルーレート イネーブルビット
RA<5:4> の各ピンに対して:
1 = ポートピンのスルーレートを制限する
0 = ポートピンのスルーレートを最大にする
- bit 3 **未実装:** 「0」として読み出し
- bit 2-0 **SLRA<2:0>:** PORTA スルーレート イネーブルビット
RA<2:0> の各ピンに対して:
1 = ポートピンのスルーレートを制限する
0 = ポートピンのスルーレートを最大にする

レジスタ 11-8: INLVLA: PORTA 入力レベル制御レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-6 **未実装:** 「0」として読み出し
- bit 5-0 **INLVLA<5:0>:** PORTA 入力レベル選択ビット
RA<5:0> の各ピンに対して:
1 = ポートピンのデジタル入力を ST しきい値で動作させる
0 = ポートピンのデジタル入力を TTL しきい値で動作させる

表 11-2: PORTA 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	142
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	140
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	141
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	139
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	142
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	141

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し。網掛けの部分は PORTA では使いません。

Note 1: 未実装、「1」として読み出します。

表 11-3: PORTA 関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ 内容記載 ページ
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

凡例: — = 未実装、「0」として読み出し。網掛けの部分は PORTA では使いません。

11.3 PORTB レジスタ (PIC16(L)F1768/9 のみ)

11.3.1 データレジスタ

PORTB は 4 ビット幅の双方向ポートです。これに対応するデータ方向レジスタが、TRISB([レジスタ 11-10](#))です。TRISB ビットをセットする (= 1) と、対応する PORTB ピンが入力になります (すなわち、対応する出力ドライバがハイ インピーダンス モードに移行します)。TRISB ビットをクリアする (= 0) と、対応する PORTB のピンが出力になります (すなわち、出力ドライバが有効になり、出力ラッチの内容が選択したピンに出力されます)。例 11-1 に、I/O ポートの初期化方法を示します。

PORTB レジスタ ([レジスタ 11-9](#)) を読み出すとピンのステータスが読み出され、PORTB レジスタに書き込むと PORT ラッチに書き込まれます。書き込み動作は全て Read-Modify-Write です。従って、ポートへの書き込み時にはまずポートピンが読み出され、この値が変更されてから PORT データラッチ (LATB) に書き込まれます。

11.3.2 方向制御

ピンをアナログ入力として使う場合も、TRISB レジスタ ([レジスタ 11-10](#)) が PORTB ピンの出力ドライバを制御します。これらのピンをアナログ入力として使う場合、必ず TRISB レジスタのビットをセットしたままにします。アナログ入力として設定された I/O ピンは常に「0」として読み出されます。

11.3.3 オープンドレイン制御

ODCONB レジスタ ([レジスタ 11-14](#)) は、ポートのオープンドレイン機能を制御します。オープンドレインの動作は、各ピンで個別に選択します。ODCONB ビットをセットすると、対応するポート出力は電流のシンクのみ可能なオープンドレイン ドライバとなります。ODCONB ビットをクリアすると、対応するポート出力ピンは電流のソースとシンクが可能な標準のプッシュプル ドライバとなります。

11.3.4 スルーレート制御

SLRCONB レジスタ ([レジスタ 11-15](#)) は、各ポートピンのスルーレートを制御します。スルーレート制御は、各ポートピンで個別に選択できます。SLRCONB ビットをセットした場合、対応するポートピンは制限されたスルーレートで駆動されます。SLRCONB ビットをクリアした場合、対応するポートピンは最大スルーレートで駆動されます。

11.3.5 入力しきい値制御

INLVLB レジスタ ([レジスタ 11-16](#)) は、各 PORTB 入力ピンの入力電圧しきい値を制御します。シュミットトリガ CMOS か TTL 互換しきい値を選択できます。入力しきい値は、PORTB レジスタの読み出し値、そして状態変化割り込み有効時には割り込みを生成するレベルを決定する重要な要素です。しきい値レベルの詳細は、[表 36-4: I/O ポート](#) を参照してください。

Note: 入力しきい値の選択は、全ての周辺モジュールを無効にした状態で変更します。アクティブなモジュールがある時にしきい値レベルを変更すると、入力ピンでの実際の電圧レベルに関係なく、そのピンに関連する状態遷移が誤って生成される場合があります。

11.3.6 アナログ制御

ANSELB レジスタ ([レジスタ 11-12](#)) は、I/O ピンの入力モードをアナログに設定する場合に使います。ANSELB ビットを High に設定したピンに対してデジタル読み出しを実行すると、全て「0」として読み出され、ピンのアナログ機能が正しく動作します。

ANSELB ビットの状態は、デジタル出力機能には影響を与えません。TRIS をクリアして ANSELB をセットしたピンはデジタル出力として動作しますが、入力モードはアナログです。このように設定されたポートに対して Read-Modify-Write 命令を実行すると予期しない挙動を引き起こす事があります。

Note: ANSELB ビットのリセット後の既定値はアナログモードです。任意のピンをデジタル汎用入力または周辺入力として使うには、対応する ANSEL ビットをユーザソフトウェアで「0」に初期化する必要があります。

11.3.7 PORTB の機能と出力の優先度

各ピンはリセット後、既定値である PORT ラッチデータに戻ります。その他の機能はペリフェラルピンセレクト ロジックで選択します。詳細は [セクション 12.0 「ペリフェラルピンセレクト \(PPS\) モジュール」](#) を参照してください。ADC およびオペアンプ入力等のアナログ入力機能はペリフェラルピンセレクトのリストには記載していません。これらの入力は、ANSELB レジスタで I/O ピンをアナログモードに設定すると有効になります。アナログモードの場合、デジタル出力機能は継続してピンを制御できます。

11.4 レジスタ定義 : PORTB

レジスタ 11-9: PORTB: PORTB レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **RB<7:4>**: PORTB 汎用 I/O ピンビット (1)

1 = PORT ピン $\geq V_{IH}$ である

0 = PORT ピン $\leq V_{IL}$ である

bit 3-0 **未実装**: 「0」として読み出し

Note 1: PORTB への書き込みは、実際には対応する LATB レジスタへの書き込み動作です。PORTB レジスタからの読み出しは、実際には I/O ピン値の読み出し動作です。

レジスタ 11-10: TRISB: PORTB 3 ステートレジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **TRISB<7:4>**: PORTB 3 ステート制御ビット

1 = PORTB ピンを入力として設定する (3 ステート)

0 = PORTB ピンを出力として設定する

bit 3-0 **未実装**: 「0」として読み出し

PIC16(L)F1764/5/8/9

レジスタ 11-11: LATB: PORTB データラッチ レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **LATB<7:4>**: PORTB 出力ラッチ値のビット⁽¹⁾

bit 3-0 **未実装**: 「0」として読み出し

Note 1: PORTB への書き込みは、実際には対応する LATB レジスタへの書き込み動作です。PORTB レジスタからの読み出しは、実際には I/O ピン値の読み出し動作です。

レジスタ 11-12: ANSELB: PORTB アナログ選択レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **ANSB<7:4>**: RB<7:4> の各ピンの機能をアナログまたはデジタルのどちらかに設定します。
0 = デジタル I/O: ピンをポートまたはデジタル特殊機能に割り当てる
1 = アナログ入力: ピンをアナログ入力として割り当て⁽¹⁾、デジタル入力バッファを無効にする

bit 3-0 **未実装**: 「0」として読み出し

Note 1: アナログ入力として設定する場合、ピンの電圧を外部から制御するには、対応する TRIS ビットをセットする必要があります。

レジスタ 11-13: WPUB: PORTB 弱プルアップ レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **WPUB<7:4>**: 弱プルアップ レジスタビット

1 = プルアップを有効にする
0 = プルアップを無効にする

bit 3-0 **未実装**: 「0」として読み出し

- Note 1:** 個別にプルアップを有効にする場合、OPTION_REG レジスタのグローバル $\overline{\text{WPUEN}}$ ビットをクリアする必要があります。
- 2:** ピンが出力として設定されている場合、弱プルアップは自動的に無効になります。

レジスタ 11-14: ODCONB: PORTB オープンドレイン制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODB7	ODB6	ODB5	ODB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **ODB<7:4>**: PORTB オープンドレイン イネーブルビット

RB<7:4> の各ピンに対して:
1 = ポートピンの動作をオープンドレイン駆動 (シンクのみ) とする
0 = ポートピンの動作を標準のプッシュプル駆動 (ソースとシンク) とする

bit 3-0 **未実装**: 「0」として読み出し

PIC16(L)F1764/5/8/9

レジスタ 11-15: SLRCONB: PORTB スルーレート制御レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **SLRB<7:4>**: PORTB スルーレート イネーブルビット
RB<7:4> の各ピンに対して:
1 = ポートピンのスルーレートを制限する
0 = ポートピンのスルーレートを最大にする

bit 3-0 **未実装**: 「0」として読み出し

レジスタ 11-16: INLVLB: PORTB 入力レベル制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **INLVLB<7:4>**: PORTB 入力レベル選択ビット
RB<7:4> の各ピンに対して:
1 = ポートピンのデジタル入力を ST しきい値で動作させる
0 = ポートピンのデジタル入力を TTL しきい値で動作させる

bit 3-0 **未実装**: 「0」として読み出し

表 11-4: PORTB 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	148
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	146
ODCONB	ODB7	ODB6	ODB5	ODB4	—	—	—	—	147
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	145
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	148
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	148
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	147

凡例: x = 未知、u = 不変、— = 未実装、「0」として読み出し。網掛けの部分は PORTB では使いません。

11.5 PORTC レジスタ

11.5.1 データレジスタ

PORTC は、PIC16(L)F1764/5 では 6 ビット幅、PIC16(L)F1768/9 では 8 ビット幅の双方向ポートです。これに対応するデータ方向レジスタが、TRISC([レジスタ 11-18](#))です。TRISC ビットをセットする (= 1) と、対応する PORTC ピンが入力になります (すなわち、対応する出力ドライバがハイ インピーダンス モードに移行します)。TRISC ビットをクリアする (= 0) と、対応する PORTC のピンが出力になります (すなわち、出力ドライバが有効になり、出力ラッチの内容が選択したピンに出力されます)。例 11-1 に、I/O ポートの初期化方法を示します。

PORTC レジスタ ([レジスタ 11-17](#)) を読み出すとピンのステータスが読み出され、PORTC レジスタに書き込むと PORT ラッチに書き込まれます。書き込み動作は全て Read-Modify-Write です。従って、ポートへの書き込み時にはまずポートピンが読み出され、この値が変更されてから PORT データラッチ (LATC) に書き込まれます。

11.5.2 方向制御

ピンをアナログ入力として使う場合も、TRISC レジスタ ([レジスタ 11-18](#)) が PORTC ピンの出力ドライバを制御します。これらのピンをアナログ入力として使う場合、必ず TRISC レジスタのビットをセットしたままにします。アナログ入力として設定された I/O ピンは常に「0」として読み出されます。

11.5.3 入力しきい値制御

INLVLC レジスタ ([レジスタ 11-24](#)) は、各 PORTC 入力ピンの入力電圧しきい値を制御します。シュミットトリガ CMOS か TTL 互換しきい値を選択できます。入力しきい値は、PORTC レジスタの読み出し値、そして状態変化割り込み有効時には割り込みを生成するレベルを決定する重要な要素です。しきい値レベルの詳細は、表 36-4: I/O ポートを参照してください。

Note: 入力しきい値の選択は、全ての周辺モジュールを無効にした状態に変更します。アクティブなモジュールがある時にしきい値レベルを変更すると、入力ピンでの実際の電圧レベルに関係なく、そのピンに関連する状態遷移が誤って生成される場合があります。

11.5.4 オープンドレイン制御

ODCONC レジスタ ([レジスタ 11-22](#)) は、ポートのオープンドレイン機能を制御します。オープンドレインの動作は、各ピンで個別に選択します。ODCONC ビットをセットすると、対応するポート出力は電流のシンクのみ可能なオープンドレイン ドライバとなります。ODCONC ビットをクリアすると、対応するポート出力ピンは電流のソースとシンクが可能な標準のプッシュプル ドライバとなります。

11.5.5 スルーレート制御

SLRCONC レジスタ ([レジスタ 11-23](#)) は、各ポートピンのスルーレートを制御します。スルーレート制御は、各ポートピンで個別に選択できます。SLRCONC ビットをセットした場合、対応するポートピンは制限されたスルーレートで駆動されます。SLRCONC ビットをクリアした場合、対応するポートピンは最大スルーレートで駆動されます。

11.5.6 アナログ制御

ANSEL レジスタ ([レジスタ 11-20](#)) は、I/O ピンの入力モードをアナログに設定する場合に使用します。ANSEL ビットを High に設定したピンに対してデジタル読み出しを実行すると、全て「0」として読み出され、ピンのアナログ機能が正しく動作します。

ANSEL ビットの状態は、デジタル出力機能には影響を与えません。TRIS をクリアして ANSEL をセットしたピンはデジタル出力として動作しますが、入力モードはアナログです。このように設定されたポートに対して Read-Modify-Write 命令を実行すると予期しない挙動を引き起こすことがあります。

Note: ANSEL ビットのリセット後の既定値はアナログモードです。任意のピンをデジタル汎用入力または周辺入力として使うには、対応する ANSEL ビットをユーザーソフトウェアで「0」に初期化する必要があります。

11.5.7 PORTC の機能と出力の優先度

各ピンはリセット後、既定値である PORT ラッチデータに戻ります。その他の機能はペリフェラル ピンセレクト ロジックで選択します。詳細は [セクション 12.0 「ペリフェラル ピンセレクト \(PPS\) モジュール」](#) を参照してください。

ADC およびコンパレータ入力等のアナログ入力機能はペリフェラル ピンセレクトのリストには記載していません。これらの入力は、ANSEL レジスタで I/O ピンをアナログモードに設定すると有効になります。アナログモードの場合、デジタル出力機能は継続してピンを制御できます。

11.5.8 高電流駆動制御

RC4 と RC5 の出力ドライバは最大 100 mA の電流をソースまたはシンクできます。この特別の駆動能力は HIDRVC レジスタ ([レジスタ 11-25](#)) の制御ビットで有効または無効にできます。

PIC16(L)F1764/5/8/9

11.6 レジスタ定義 : PORTC

レジスタ 11-17: PORTC: PORTC レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7 ⁽²⁾	RC6 ⁽²⁾	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **RC<7:0>**: PORTC 汎用 I/O ピンビット (1, 2)

1 = PORT ピン $\geq V_{IH}$ である

0 = PORT ピン $\leq V_{IL}$ である

Note 1: PORTC への書き込みは、実際には対応する LATC レジスタへの書き込み動作です。PORTC レジスタからの読み出しは、実際には I/O ピン値の読み出し動作です。

2: RC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-18: TRISC: PORTC 3 ステートレジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **TRISC<7:0>**: PORTC 3 ステート制御ビット (1)

1 = PORTC ピンを入力として設定する (3 ステート)

0 = PORTC ピンを出力として設定する

Note 1: TRISC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-19: LATC: PORTC データラッチ レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **LATC<7:0>**: PORTC 出カラッチ値のビット⁽¹⁾

Note 1: LATC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-20: ANSEL: PORTC アナログ選択レジスタ

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **ANSC<7:0>**: RC<7:6> の各ピンの機能をアナログまたはデジタルのどちらかに設定します。^(1,2)
1 = アナログ入力: ピンをアナログ入力として割り当て⁽¹⁾、デジタル入力バッファを無効にする
0 = デジタル I/O: ピンをポートまたはデジタル特殊機能に割り当てる

bit 5-4 **未実装**: 「0」として読み出し

bit 3-0 **ANSC<3:0>**: RC<3:0> の各ピンの機能をアナログまたはデジタルのどちらかに設定します。⁽¹⁾
1 = アナログ入力: ピンをアナログ入力として割り当て⁽¹⁾、デジタル入力バッファを無効にする
0 = デジタル I/O: ピンをポートまたはデジタル特殊機能に割り当てる

Note 1: アナログ入力として設定する場合、ピンの電圧を外部から制御するには、対応する TRIS ビットをセットする必要があります。

2: ANSC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

PIC16(L)F1764/5/8/9

レジスタ 11-21: WPUC: PORTC 弱プルアップ レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUC7 ⁽³⁾	WPUC6 ⁽³⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **WPUC<7:0>**: 弱プルアップ レジスタビット ⁽¹⁾、⁽²⁾
1 = プルアップを有効にする
0 = プルアップを無効にする

- Note 1:** 個別にプルアップを有効にする場合、OPTION_REG レジスタのグローバル $\overline{\text{WPUEN}}$ ビットをクリアする必要があります。
2: ピンが出力として設定されている場合、弱プルアップは自動的に無効になります。
3: WPUC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-22: ODCONC: PORTC オープンドレイン制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **ODC<7:0>**: PORTC オープンドレイン イネーブルビット ⁽¹⁾
RC<7:0> の各ピンに対して:
1 = ポートピンの動作をオープンドレイン駆動 (シンクのみ) とする
0 = ポートピンの動作を標準のプッシュプル駆動 (ソースとシンク) とする

- Note 1:** ODC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-23: SLRCONC: PORTC スルーレート制御レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **SLRC<7:0>**: PORTC スルーレート イネーブルビット ⁽¹⁾
RC<7:0> の各ピンに対して:
1 = ポートピンのスルーレートを制限する
0 = ポートピンのスルーレートを最大にする

Note 1: SLRC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

レジスタ 11-24: INLVLC: PORTC 入力レベル制御レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **INLVLC<7:0>**: PORTC 入力レベル選択ビット ⁽¹⁾
RC<7:0> の各ピンに対して:
1 = ポートピンのデジタル入力を ST しきい値で動作させる
0 = ポートピンのデジタル入力を TTL しきい値で動作させる

Note 1: INLVLC<7:6> は、PIC16(L)F1768/9 でのみ使用可能です。

PIC16(L)F1764/5/8/9

レジスタ 11-25: HIDRVC: PORTC 大電流駆動制御レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	HIDC5	HIDC4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-4 **HIDC<5:4>:** PORTC 大電流駆動イネーブルビット
 RC<5:4> の各ピンに対して:
 1 = 大電流ソースおよびシンクを有効にする
 0 = 標準の電流ソースおよびシンク
bit 3-0 **未実装:** 「0」として読み出し

表 11-5: PORTC 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ内容記載ページ
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	154
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	153
LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	151
ODCONC	ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	152
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	150
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	153
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	152

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し。網掛けの部分は PORTC では使いません。

Note 1: PIC16(L)F1768/9 のみです。

12.0 ペリフェラルピンセレクト (PPS) モジュール

ペリフェラルピンセレクト(PPS)モジュールは周辺モジュールの入出力をデバイスのI/Oピンに接続します。割り当てを変更できるのはデジタル信号のみです。アナログ入出力の割り当ては変更できません。概略ブロック図図 12-1 に示すように、入力と出力は別々に選択します。

12.1 PPS 入力

各周辺モジュールには、入力を選択するために使う PPS レジスタがあります。入力にはデバイスピンが含まれます。

複数の周辺モジュールが同じ入力源で同時に動作できます。ポート読み出しは、周辺モジュールの PPS 選択には関係なく、常にピンレベルを返します。ピンがアナログ機能にも対応している場合、デジタル入力バッファを有効にするには、そのピンの ANSEL ビットをクリアする必要があります。

各周辺モジュールには専用の PPS 入力選択レジスタがありますが、レジスタ 12-1 に示すように、選択方法は全モジュールで同じです。

Note: レジスタ名の「xxx」という表記は、周辺モジュール識別子のプレースホルダです。例：CLC1PPS

12.2 PPS 出力

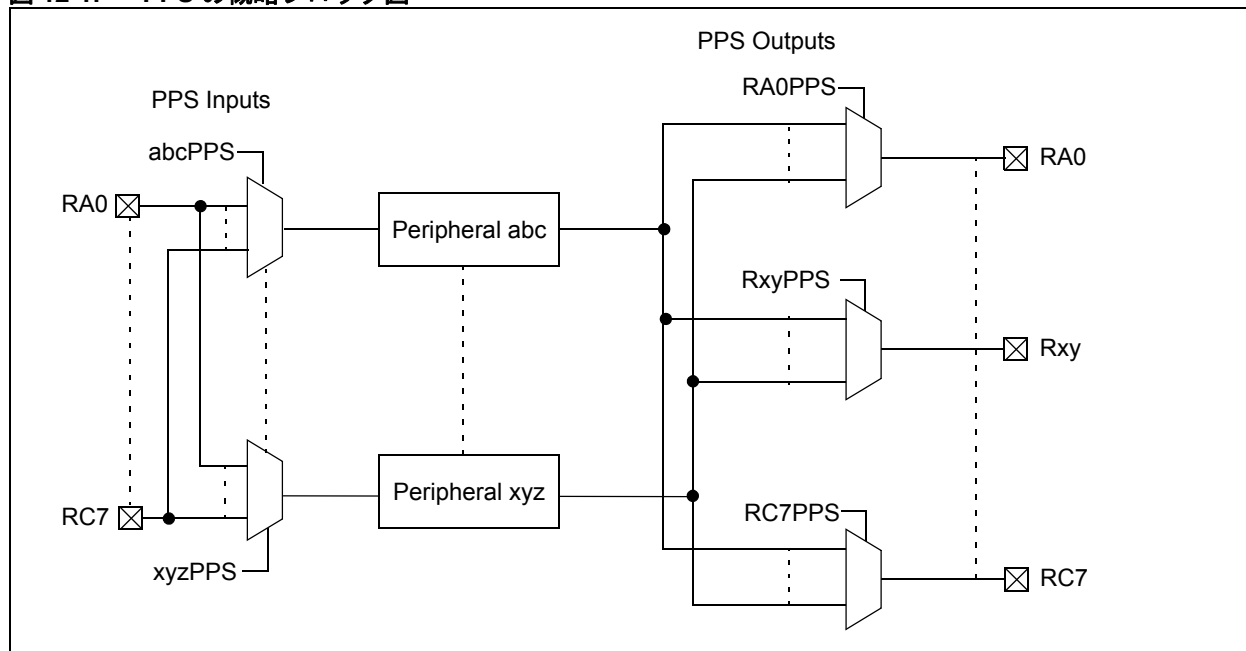
各 I/O ピンには、ピンの出力源を選択するために使う PPS レジスタがあります。基本的に、ピンの出力ドライバはそのピンに対応する TRISx レジスタで制御します。ただし一部の周辺モジュールが出力ドライバを制御する場合、必要に応じて TRISx レジスタではなく周辺モジュール自身が出力ドライバを制御します。そのような周辺モジュールには以下のものがあります。

- EUSART(同期モード)
- MSSP(I²C)
- COG(自動シャットダウン)

全てのピンに別々の PPS 周辺モジュール選択レジスタがあっても、レジスタ 12-2 に示すように、選択方法は全ピンで同じです。

Note: 「Rxy」という表記は、ピン識別子のプレースホルダです。例：RA0PPS

図 12-1: PPS の概略ブロック図



12.3 双方向ピン

同じピンで双方向通信を行う周辺モジュールの場合、PPS 入力と PPS 出力で同じピンを選択する必要があります。双方向通信を行う周辺モジュールには以下のものがあります。

- EUSART(同期モード)
- MSSP (I²C)

Note: I²C 既定値入力ピンは、I²C および SMBus と互換であり、デバイスでこの互換性を備える唯一のピンです。

12.4 PPS ロック

PPS には、意図しない変更を防ぐために全入出力をロックできるモードがあります。PPS の設定をロックするには、PPSLOCK レジスタの PPSLOCKED ビットをセットします。このビットをセットまたはクリアするには、意図しない変更を防ぐために特別なシーケンスが必要です。例 12-1 に、PPSLOCKED ビットのセットとクリアの例を示します。

例 12-1: PPS ロック / 解除シーケンス

```
; suspend interrupts
    bcf    INTCON,GIE
; BANKSEL PPSLOCK    ; set bank
; required sequence, next 5 instructions
    movlw 0x55
    movwf PPSLOCK
    movlw 0xAA
    movwf PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    bsf    PPSLOCK,PPSLOCKED
; restore interrupts
    bsf    INTCON,GIE
```

12.5 PPS の永久ロック

PPS1WAY コンフィグレーション ビットをセットすると、PPS を恒久的にロックできます。このビットをセットすると、PPSLOCKED ビットはデバイスリセット後に 1 回しかクリアまたはセットできません。これにより、PPSLOCKED ビットをクリアして初期化中に入出力を選択できます。全ての入出力を設定した後で PPSLOCKED ビットをセットすると、このビットはセットされたままとなり、次回デバイスをリセットするまでクリアできません。

12.6 スリープ中の動作

PPSによる入出力設定はスリープの影響を受けません。

12.7 リセットの影響

デバイスのパワーオン リセット (POR) は PPS の全入出力設定をクリアし、既定値に戻します。その他の全てのリセットでは、PPS の設定は変更されません。表 12-1 に入力選択の既定値を示します。

PIC16(L)F1764/5/8/9

12.8 レジスタ定義 : PPS 入力選択

レジスタ 12-1: xxxPPS: 周辺モジュール xxx の入力選択

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	
—	—	—	xxxPPS<4:0>					
bit 7							bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 周辺モジュールにより異なる

bit 7-5 **未実装:** 「0」として読み出し

bit 4-3 **xxxPPS<4:3>:** 周辺モジュール xxx の入力 PORT 選択ビット

11 = 予約済み使用不可
10 = 周辺モジュールの入力を PORTC とする
01 = 周辺モジュールの入力を PORTB とする (2)
00 = 周辺モジュールの入力を PORTA とする

bit 2-0 **xxxPPS<2:0>:** 周辺モジュール xxx の入力ビット選択ビット (1)

111 = 周辺モジュールの入力を PORTx の Bit 7 とする (Rx7)
110 = 周辺モジュールの入力を PORTx の Bit 6 とする (Rx6)
101 = 周辺モジュールの入力を PORTx の Bit 5 とする (Rx5)
100 = 周辺モジュールの入力を PORTx の Bit 4 とする (Rx4)
011 = 周辺モジュールの入力を PORTx の Bit 3 とする (Rx3)
010 = 周辺モジュールの入力を PORTx の Bit 2 とする (Rx2)
001 = 周辺モジュールの入力を PORTx の Bit 1 とする (Rx1)
000 = 周辺モジュールの入力を PORTx の Bit 0 とする (Rx0)

Note 1: xxxPPS レジスタの一覧とリセット値は表 12-1 を参照してください。

2: PIC16(L)F1768/9 のみです。

レジスタ 12-2: RxyPPS: Rxy ピン出力源選択レジスタ

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	
—	—	—	RxyPPS<4:0>					
bit 7							bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-5 **未実装:** 「0」として読み出し

bit 4-0 **RxyPPS<4:0>:** Rxy ピン出力源選択ビット

ポートピンの出力信号を選択されたコード値で決定します。
表 12-2 に選択コード値を示します。

レジスタ 12-3: PPSLOCK: PPS ロックレジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-1 **未実装:** 「0」として読み出し

bit 0 **PPSLOCKED:** PPS ロックビット

1 = PPS をロックする。PPS の選択は変更できない

0 = PPS をロックしない。PPS の選択は変更できる

PIC16(L)F1764/5/8/9

表 12-1: PPS 入力レジスタのリセット値

周辺モジュール	xxxPPS レジスタ	ピン選択の既定値		リセット値 (xxxPPS<4:0>)	
		PIC16(L)F1768/9	PIC16(L)F1764/5	PIC16(L)F1768/9	PIC16(L)F1764/5
状態変化割り込み	INTPPS	RA2	RA2	00010	00010
Timer0 クロック	T0CKIPPS	RA2	RA2	00010	00010
Timer1 クロック	T1CKIPPS	RA5	RA5	00101	00101
Timer1 ゲート	T1GPPS	RA4	RA4	00100	00100
Timer2 クロック	T2CKIPPS	RA5	RA5	0101	0101
Timer3 クロック	T3CKIPPS	RC5	RC5	10101	10101
Timer3 ゲート	T3GPPS	RC4	RC4	10100	10100
Timer4 クロック	T4CKIPPS	RC1	RC1	10001	10001
Timer5 クロック	T5CKIPPS	RC0	RC0	10000	10000
Timer5 ゲート	T5GPPS	RC3	RC3	10011	10011
Timer6 クロック	T6CKIPPS	RA3	RA3	00011	00011
CCP1	CCP1PPS	RC5	RC5	10101	10101
CCP2	CCP2PPS ⁽¹⁾	RC3	—	10011	—
COG1	COG1INPPS	RA2	RA2	00010	00010
COG2	COG2INPPS ⁽¹⁾	RA2	—	00010	—
SPI および I ² C クロック	SSPCLKPPS	RB6	RC0	01110	10000
SPI および I ² C データ	SSPDATPPS	RB4	RC1	01100	10001
SPI スレーブ選択	SSPSSPPS	RC6	RC3	10110	10011
EUSART 受信	RXPPS	RB5	RC5	01101	10101
EUSART クロック	CKPPS	RB7	RC4	01111	10100
全ての CLC	CLCIN0PPS	RC3	RC3	10011	10011
全ての CLC	CLCIN1PPS	RC4	RC4	10100	10100
全ての CLC	CLCIN2PPS	RC1	RC1	10001	10001
全ての CLC	CLCIN3PPS	RA5	RA5	00101	00101
PRG1 set_rising	PRG1RPPS	RC4	RC4	10100	10100
PRG1 set_falling	PRG1FPPS	RC5	RC5	10101	10101
PRG2 set_rising	PRG2RPPS ⁽¹⁾	RC4	—	10100	—
PRG2 set_falling	PRG2FPPS ⁽¹⁾	RC5	—	10101	—
DSM1 搬送波 High	MD1CHPPS	RA3	RA3	00011	00011
DSM1 搬送波 Low	MD1CLPPS	RA4	RA4	00100	00100
DSM1 変調	MD1MODPPS	RA5	RA5	00101	00101
DSM2 搬送波 High	MD2CHPPS ⁽¹⁾	RA3	—	00011	—
DSM2 搬送波 Low	MD2CLPPS ⁽¹⁾	RA4	—	00100	—
DSM2 変調	MD2MODPPS ⁽¹⁾	RA5	—	00101	—

例: CCP1PPS = 0x13 の場合、CCP1 入力として RC3 を選択します。

Note 1: PIC16(L)F1768/9 の場合のみです。

PIC16(L)F1764/5/8/9

表 12-2: 各周辺モジュールの出力に利用可能なポート (2)

RxyPPS<4:0>	出力信号	PIC16(L)F1768/9			PIC16(L)F1764/5	
		PORTA	PORTB	PORTC	PORTA	PORTC
1111x	予約済み	—	—	—	—	—
11101	MD2_out	●	●	●	—	—
11100	MD1_out	●	●	●	●	●
11011	sync_C4OUT	●	●	●	—	—
11010	sync_C3OUT	●	●	●	—	—
11001	sync_C2OUT	●	●	●	●	●
11000	sync_C1OUT	●	●	●	●	●
10111	DT ⁽¹⁾	●	●	●	●	●
10110	TX/CK ⁽¹⁾	●	●	●	●	●
10101	予約済み	—	—	—	—	—
10100	SDO	●	●	●	●	●
10011	SDA	●	●	●	●	●
10010	SCK/SCL ⁽¹⁾	●	●	●	●	●
10001	PWM6_out	●	●	●	—	—
10000	PWM5_out	●	●	●	●	●
01111	PWM4_out	●	●	●	—	—
01110	PWM3_out	●	●	●	●	●
01101	CCP2_out	●	●	●	●	●
01100	CCP1_out	●	●	●	●	●
01011	COG2D ⁽¹⁾	●	●	●	—	—
01010	COG2C ⁽¹⁾	●	●	●	—	—
01001	COG2B ⁽¹⁾	●	●	●	—	—
01000	COG2A ⁽¹⁾	●	●	●	—	—
00111	COG1D ⁽¹⁾	●	●	●	●	●
00110	COG1C ⁽¹⁾	●	●	●	●	●
00101	COG1B ⁽¹⁾	●	●	●	●	●
00100	COG1A ⁽¹⁾	●	●	●	●	●
00011	LC3_out	●	●	●	●	●
00010	LC2_out	●	●	●	●	●
00001	LC1_out	●	●	●	●	●
00000	LATxy	●	●	●	●	●

Note 1: 必要に応じて、TRIS 制御より周辺モジュールによる制御が優先されます。

2: サポートされない周辺モジュールは「0」を出力します。

PIC16(L)F1764/5/8/9

表 12-3: PPS モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	159
INTPPS	—	—	—	INTPPS<4:0>					158
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					158
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					158
T1GPPS	—	—	—	T1GPPS<4:0>					158
T2CKIPPS	—	—	—	T2CKIPPS<4:0>					158
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					158
T3GPPS	—	—	—	T3GPPS<4:0>					158
T4CKIPPS	—	—	—	T4CKIPPS<4:0>					158
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					158
T5GPPS	—	—	—	T5GPPS<4:0>					158
T6CKIPPS	—	—	—	T6CKIPPS<4:0>					158
CCP1PPS	—	—	—	CCP1PPS<4:0>					158
CCP2PPS ⁽¹⁾	—	—	—	CCP2PPS<4:0>					158
COG1INPPS	—	—	—	COG1INPPS<4:0>					158
COG2INPPS ⁽¹⁾	—	—	—	COG2INPPS<4:0>					158
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					158
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					158
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					158
RXPPS	—	—	—	RXPPS<4:0>					158
CKPPS	—	—	—	CKPPS<4:0>					158
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					158
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					158
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					158
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					158
PRG1RPPS	—	—	—	PRG1RPPS<4:0>					158
PRG1FPPS	—	—	—	PRG1FPPS<4:0>					158
PRG2RPPS ⁽¹⁾	—	—	—	PRG2RPPS<4:0>					158
PRG2FPPS ⁽¹⁾	—	—	—	PRG2FPPS<4:0>					158
MD1CHPPS	—	—	—	MD1CHPPS<4:0>					158
MD1CLPPS	—	—	—	MD1CLPPS<4:0>					158
MD1MODPPS	—	—	—	MD1MODPPS<4:0>					158
MD2CHPPS ⁽¹⁾	—	—	—	MD2CHPPS<4:0>					158
MD2CLPPS ⁽¹⁾	—	—	—	MD2CLPPS<4:0>					158
MD2MODPPS ⁽¹⁾	—	—	—	MD2MODPPS<4:0>					158
RA0PPS	—	—	—	RA0PPS<4:0>					158
RA1PPS	—	—	—	RA1PPS<4:0>					158
RA2PPS	—	—	—	RA2PPS<4:0>					158
RA4PPS	—	—	—	RA4PPS<4:0>					158
RA5PPS	—	—	—	RA5PPS<4:0>					158
RB4PPS ⁽¹⁾	—	—	—	RB4PPS<4:0>					158

表 12-3: PPS モジュール関連レジスタのまとめ (続き)

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
RB5PPS ⁽¹⁾	—	—	—	RB5PPS<4:0>					158
RB6PPS ⁽¹⁾	—	—	—	RB6PPS<4:0>					158
RB7PPS ⁽¹⁾	—	—	—	RB7PPS<4:0>					158
RC0PPS	—	—	—	RC0PPS<4:0>					158
RC1PPS	—	—	—	RC1PPS<4:0>					158
RC2PPS	—	—	—	RC2PPS<4:0>					158
RC3PPS	—	—	—	RC3PPS<4:0>					158
RC4PPS	—	—	—	RC4PPS<4:0>					158
RC5PPS	—	—	—	RC5PPS<4:0>					158
RC6PPS ⁽¹⁾	—	—	—	RC6PPS<4:0>					158
RC7PPS ⁽¹⁾	—	—	—	RC7PPS<4:0>					158

凡例: — = 未実装、「0」として読み出し。網掛けの部分は PPS モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

13.0 状態変化割り込み

全ポートの全ピンは状態変化割り込み (IOC) ピンとして設定できます。割り込みは、信号の立ち上がりエッジ/立ち下がりエッジのどちらでも生成できます。1本のピンでも複数のピンの組み合わせでも割り込み生成が可能です。状態変化割り込みモジュールには、以下の特長があります。

- 状態変化割り込みイネーブル (マスタスイッチ)
- 個別のピン設定
- 立ち上がり/立ち下がりエッジの検出
- 個別のピン割り込みフラグ

図 13-1 に、IOC モジュールのブロック図を示します。

13.1 モジュールの有効化

各ピンで割り込みを生成するには、INTCON レジスタの IOCIE ビットをセットする必要があります。IOCIE ビットがクリアの場合、ピンにおいてエッジは検出されますが割り込みは生成されません。

13.2 個別のピン設定

各ピンには、立ち上がりエッジと立ち下がりエッジの検出機能があります。立ち上がりエッジの検出機能を有効にする場合、IOCxP レジスタの対応するビットをセットします。立ち下がりエッジの検出機能を有効にする場合、IOCxN レジスタの対応するビットをセットします。

IOCxP および IOCxN レジスタ両方で対応するビットをセットすると、立ち上がりエッジと立ち下がりエッジ両方を検出できます。

13.3 割り込みフラグ

IOCxF レジスタのビットは、各ポートの状態変化割り込みピンに対応するステータスフラグです。状態変化割り込みを有効にしたピンに指定したエッジが検出されると、そのピンのステータスフラグがセットされ、IOCIE ビットをセットしている場合は割り込みが生成されます。INTCON レジスタの IOCIF ビットは、全ての IOCxF ビットのステータスを反映します。

13.4 割り込みフラグのクリア

各ステータスフラグ (IOCxF レジスタビット) は、ゼロにリセットする事でクリアできます。クリア動作中に次のエッジが検出された場合、実際に書き込まれる値に関係なく、クリア動作完了時に対応するステータスフラグがセットされます。

フラグクリア中に検出したエッジを失わないようにするため、既知の変更済みビットをマスクした AND 動作のみ実行します。以下に、そのシーケンスの例を示します。

例 13-1: 割り込みフラグのクリア (PORTA の例)

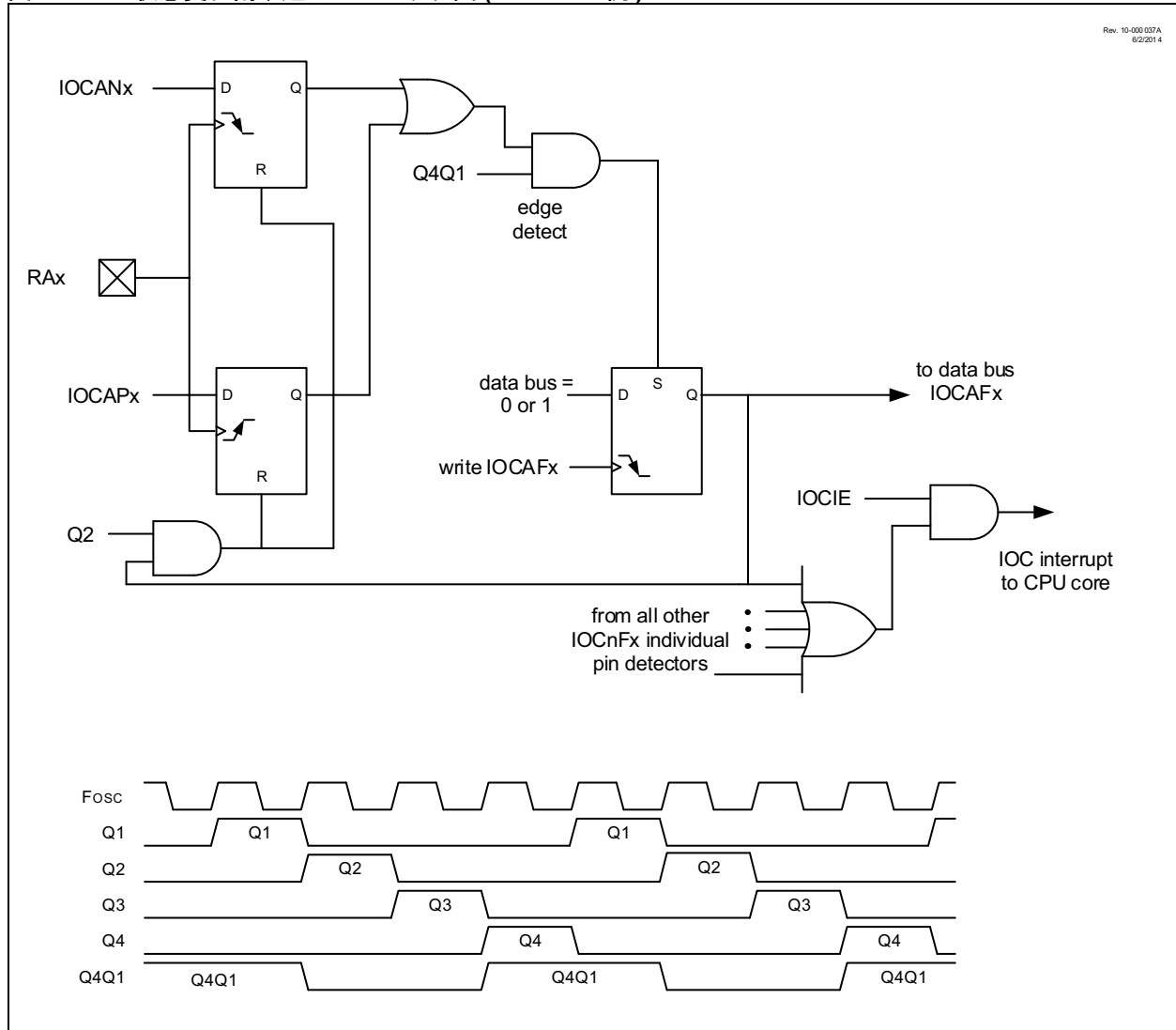
```
MOVLW 0xff
XORWF IOCAF, W
ANDWF IOCAF, F
```

13.5 スリープ中の動作

IOCIE ビットがセットされている場合、状態変化割り込みシーケンスはデバイスをスリープから復帰させます。

スリープ中にエッジが検出された場合、対応する IOCxF レジスタが更新された後に復帰後の最初の命令が実行されます。

図 13-1: 状態変化割り込みのブロック図 (PORTA の例)



PIC16(L)F1764/5/8/9

13.6 レジスタ定義：状態変化割り込み制御

レジスタ 13-1: IOCAP: PORTA 立ち上がりエッジ状態変化割り込みレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7							bit 0

凡例：

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装**：「0」として読み出し

bit 5-0 **IOCAP<5:0>**: PORTA 立ち上がりエッジ状態変化割り込みイネーブルビット
1 = 立ち上がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCAF_x ビットと IOCIF フラグがセットされる
0 = 対応するピンの状態変化割り込みを無効にする

レジスタ 13-2: IOCAN: PORTA 立ち下がりエッジ状態変化割り込みレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
bit 7							bit 0

凡例：

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装**：「0」として読み出し

bit 5-0 **IOCAN<5:0>**: PORTA 立ち下がりエッジ状態変化割り込みイネーブルビット
1 = 立ち下がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCAF_x ビットと IOCIF フラグがセットされる
0 = 対応するピンの状態変化割り込みを無効にする

レジスタ 13-3: IOCAF: PORTA 状態変化割り込みフラグレジスタ

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア HS = ビットはハードウェアでセット

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **IOCAF<5:0>:** PORTA 状態変化割り込みフラグビット

1 = 対応するピンで状態変化が検出された。

IOCAPx = 1 で、RAX に立ち上がりエッジが検出された場合、または IOCANx = 1 で、RAX に立ち下がりエッジが検出された場合にセットされる

0 = 状態変化なし、または検出された状態変化をユーザがクリアした

レジスタ 13-4: IOCBP: PORTB 立ち上がりエッジ状態変化割り込みレジスタ⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **IOCBP<7:4>:** PORTB 立ち上がりエッジ状態変化割り込みイネーブルビット

1 = 立ち上がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCBFx ビットと IOCIF フラグがセットされる

0 = 対応するピンの状態変化割り込みを無効にする

bit 3-0 **未実装:** 「0」として読み出し

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 13-5: IOCBN: PORTB 立ち下がりエッジ状態変化割り込みレジスタ ⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **IOCBN<7:4>:** PORTB 立ち下がりエッジ状態変化割り込みイネーブルビット
1 = 立ち下がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCBFx ビットと IOCIF フラグがセットされる
0 = 対応するピンの状態変化割り込みを無効にする

bit 3-0 **未実装:** 「0」として読み出し

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 13-6: IOCBF: PORTB 状態変化割り込みフラグレジスタ ⁽¹⁾

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア HS = ビットはハードウェアでセット

bit 7-4 **IOCBF<7:4>:** PORTB 状態変化割り込みフラグビット
1 = 対応するピンで状態変化が検出された。
IOCBPx = 1 の時 RBx で立ち上がりエッジが検出された場合、または IOCBNx = 1 の時 RBx で立ち下がりエッジが検出された場合にセットされます。
0 = 状態変化なし、または検出された状態変化をユーザがクリアした

bit 3-0 **未実装:** 「0」として読み出し

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 13-7: IOCCP: PORTC 立ち上がりエッジ状態変化割り込みレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **IOCCP<7:0>**: PORTC 立ち上がりエッジ状態変化割り込みイネーブルビット
1 = 立ち上がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCCFx ビットと IOCIF フラグがセットされる
0 = 対応するピンの状態変化割り込みを無効にする

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 13-8: IOCCN: PORTC 立ち下がりエッジ状態変化割り込みレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **IOCCN<7:0>**: PORTC 立ち下がりエッジ状態変化割り込みイネーブルビット
1 = 立ち下がりエッジの状態変化割り込みを有効にする。エッジ検出時に IOCCFx ビットと IOCIF フラグがセットされる
0 = 対応するピンの状態変化割り込みを無効にする

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 13-9: IOCCF: PORTC 状態変化割り込みフラグレジスタ

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア HS = ビットはハードウェアでセット

bit 7-0 **IOCCF<7:0>**: PORTC 状態変化割り込みフラグビット
1 = 対応するピンで状態変化が検出された。
IOCCPx = 1 の時 RCx で立ち上がりエッジが検出された場合、または IOCCNx = 1 の時 RCx で立ち下がりエッジが検出された場合にセットされます。
0 = 状態変化なし、または検出された状態変化をユーザがクリアした

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

表 13-1: 状態変化割り込み関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	167
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	166
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	166
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	168
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	168
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	167
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	169
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	169
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	169
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例: — = 未実装、「0」として読み出し。網掛けの部分は状態変化割り込みでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

14.0 固定参照電圧 (FVR)

固定参照電圧 (FVR) は、VDD から独立している安定した参照電圧であり、出力レベルを 1.024 V、2.048 V、4.096 V のいずれかから選択できます。FVR の出力は、設定によって以下の参照電圧として供給できます。

- ADC 入力チャンネル
- ADC 正側参照電圧
- コンパレータ正入力
- D/A コンバータ (DAC)

FVRCON レジスタの FVREN ビットをセットすると FVR を有効にできます。

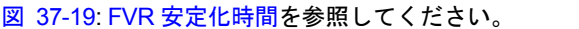
14.1 独立ゲインアンプ

FVR の出力は、2 つの独立したプログラマブル ゲインアンプを介して ADC、コンパレータ、DAC へ供給されます。各アンプはゲインを 1 倍、2 倍、4 倍のいずれかにするようプログラミングでき、これによって電圧レベルを 3 種類に切り換える事ができます。

ADC モジュールへ供給する参照電圧のゲインアンプを有効化 / 設定するには、FVRCON レジスタの ADFVR<1:0> ビットを使います。詳細は、[セクション 16.0 「A/D コンバータ \(ADC\) モジュール」](#) を参照してください。

DAC およびコンパレータ モジュールへ供給する参照電圧のゲインアンプを有効化/設定するには、FVRCON レジスタの CDAFVR<1:0> ビットを使います。詳細は、[セクション 17.0 「5 ビット D/A コンバータ \(DAC\) モジュール」](#) と [セクション 19.0 「コンパレータ モジュール」](#) を参照してください。

14.2 FVR 安定化時間

固定参照電圧モジュールを有効にした場合、参照電圧回路とアンプ回路が安定するまである程度の時間が必要です。回路が安定して使える状態になると、FVRCON レジスタの FVRRDY ビットがセットされます。
 [図 37-19: FVR 安定化時間](#) を参照してください。

14.3 FVR バッファ安定化時間

FVR バッファ 1 または 2 のどちらかを ON にすると、バッファアンプ回路が安定化するのに 30 us 必要となります。この安定化時間は FVR が既に動作中で安定化している場合でも必要です。

PIC16(L)F1764/5/8/9

図 14-1: 参照電圧のブロック図

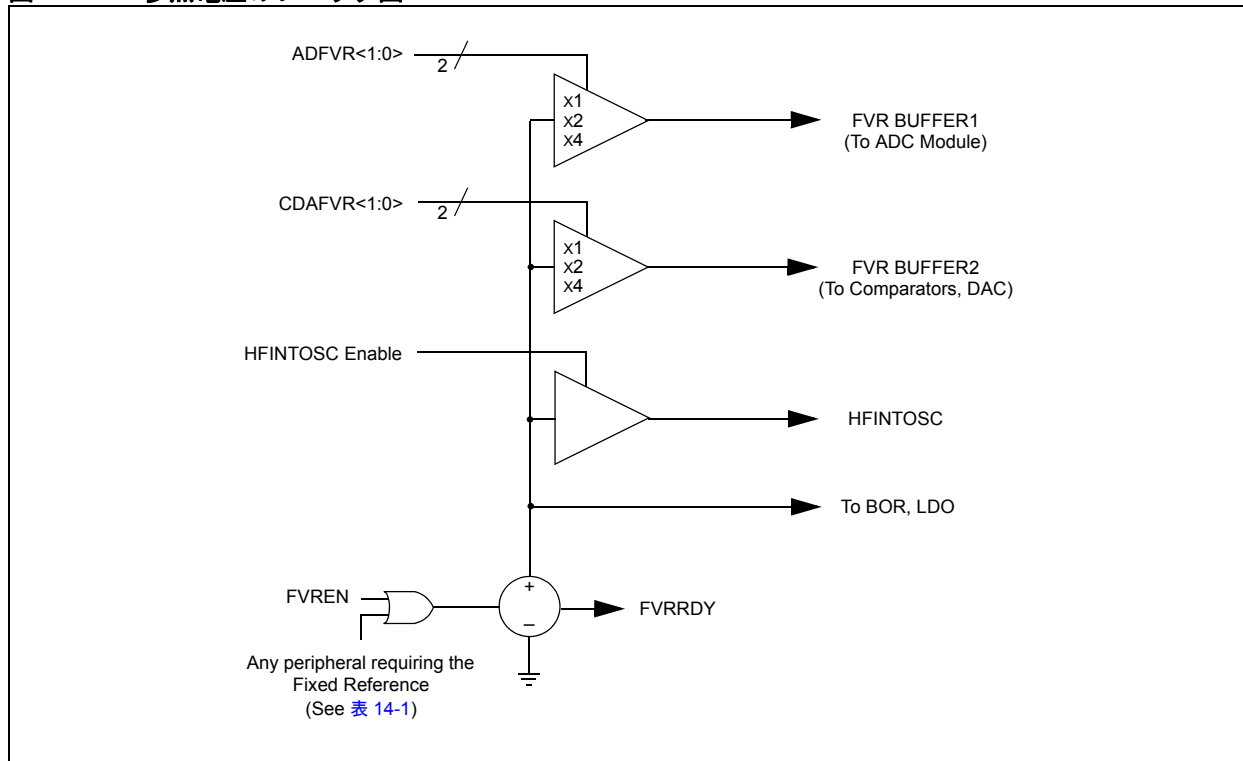


表 14-1: 固定参照電圧 (FVR) を必要とする周辺モジュール

周辺モジュール	条件	説明
HFINTOSC	FOSC<2:0> = 100 かつ IRCF<3:0> ≠ 000x	INTOSC が有効でデバイスがスリープ状態ではない
BOR	BOREN<1:0> = 11	BOR を常に有効にする
	BOREN<1:0> = 10 かつ BORFS = 1	スリープ中に BOR を無効にして、BOR 高速起動を有効にする
	BOREN<1:0> = 01 かつ BORFS = 1	BOR をソフトウェアで制御し、BOR 高速起動を有効にする
LDO	全ての PIC16F1764/5/8/9 で VREGPM = 1 かつスリープ中でない 場合	スリープ中、デバイスは ULP レギュレータで動作する

14.4 レジスタ定義 : FVR 制御

レジスタ 14-1: FVRCON: 固定参照電圧制御レジスタ

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY ⁽¹⁾	TSEN ⁽³⁾	TSRNG ⁽³⁾	CDAFVR<1:0>		ADFVR<1:0>	
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7	FVREN: 固定参照電圧イネーブルビット 1 = 固定参照電圧を有効にする 0 = 固定参照電圧を無効にする
bit 6	FVRRDY: 固定参照電圧レディ フラグビット ⁽¹⁾ 1 = 固定参照電圧出力は使える状態である 0 = 固定参照電圧出力の準備が整っていない (または無効である)
bit 5	TSEN: 温度インジケータ イネーブルビット ⁽³⁾ 1 = 温度インジケータを有効にする 0 = 温度インジケータを無効にする
bit 4	TSRNG: 温度インジケータ レンジ選択ビット ⁽³⁾ 1 = VOUT = VDD - 4VT (High レンジ) 0 = VOUT = VDD - 2VT (Low レンジ)
bit 3-2	CDAFVR<1:0>: コンパレータ FVR バッファゲイン選択ビット 11 = コンパレータ FVR バッファゲインは 4 倍、出力 VCDAFVR = 4x VFVR ⁽²⁾ 10 = コンパレータ FVR バッファゲインは 2 倍、出力 VCDAFVR = 2x VFVR ⁽²⁾ 01 = コンパレータ FVR バッファゲインは 1 倍、出力 VCDAFVR = 1x VFVR 00 = コンパレータ FVR バッファは OFF
bit 1-0	ADFVR<1:0>: ADC FVR バッファゲイン選択ビット 11 = ADC FVR バッファゲインは 4 倍、出力 VADFVR = 4x VFVR ⁽²⁾ 10 = ADC FVR バッファゲインは 2 倍、出力 VADFVR = 2x VFVR ⁽²⁾ 01 = ADC FVR バッファゲインは 1 倍、出力 VADFVR = 1x VFVR 00 = ADC FVR バッファは OFF

- Note 1:** FVRRDY は、PIC16F1764/5/8/9 の場合のみ、常に「1」として読み出されます。
2: 固定参照電圧出力は、VDD を超える事ができません。
3: 詳細は [セクション 15.0「温度インジケータ モジュール」](#) を参照してください。

表 14-2: 固定参照電圧関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173

凡例: 網掛け部分は固定参照電圧では使いません。

PIC16(L)F1764/5/8/9

15.0 温度インジケータ モジュール

このデバイスファミリは、シリコンダイの動作温度を計測するために設計された温度回路を搭載しています。この回路の動作温度レンジは $-40 \sim +85 \text{ }^\circ\text{C}$ で、出力はシリコンダイの温度に比例した電圧です。温度インジケータの出力は、デバイス内蔵 ADC に接続されています。

この回路は校正方法に応じて温度しきい値の検出器、またはより高精度な温度計として使えます。1 点校正を行うと、回路はその前後の温度を計測できます。2 点校正を行えば、全温度レンジでより高精度の温度検出が可能です。校正手順の詳細は、アプリケーションノート AN1333『内部温度インジケータの使用と校正』(DS01333) を参照してください。

15.1 回路動作

図 15-1 に、温度回路の概略ブロック図を示します。温度に比例する電圧出力は、複数のシリコン接合の順方向電圧降下を計測する事で得られます。

式 15-1 は、温度インジケータの出力特性を表すものです。

式 15-1: V_{OUT} のレンジ

High レンジ: $V_{OUT} = V_{DD} - 4V_T$

Low レンジ: $V_{OUT} = V_{DD} - 2V_T$

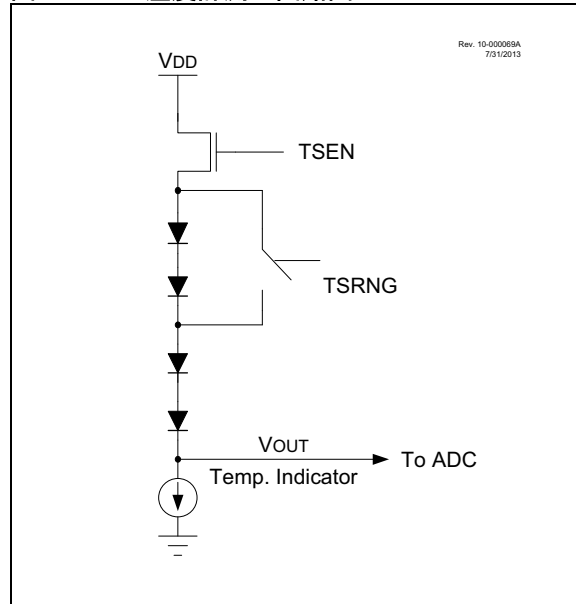
温度検出回路は、固定参照電圧 (FVR) モジュールと統合されています。詳細は [セクション 14.0 「固定参照電圧 \(FVR\)」](#) を参照してください。

この回路は、FVRCON レジスタの TSEN ビットをセットすると有効になります。無効にした場合、回路には電流が一切流れません。

回路は High レンジまたは Low レンジで動作します。FVRCON レジスタの TSRNG ビットをセットすると High レンジが選択され、より広い出力電圧が得られます。この場合、仕様温度全体にわたって分解能が向上しますが、製品間のばらつきが大きくなります。High レンジの動作には、より高いバイアス電圧が必要なため、 V_{DD} を高くする必要があります。

FVRCON レジスタの TSRNG ビットをクリアすると Low レンジが選択されます。Low レンジでは発生する電圧降下が小さいため、回路動作にはより低いバイアス電圧が必要です。Low レンジは、低電圧動作向けに用意されたモードです。

図 15-1: 温度計測の回路図



15.2 最小動作 V_{DD}

温度回路を Low レンジで動作させる場合、デバイスは仕様内の任意の動作電圧で動作させる事ができます。

一方、High レンジで動作させる場合、温度回路に適切なバイアスを印加するのに十分な V_{DD} でデバイスを動作させる必要があります。

表 15-1 に、各レンジ設定に対する最小 V_{DD} の推奨値を示します。

表 15-1: 各レンジに対する推奨 V_{DD}

最小 V_{DD} , TSRNG = 1	最小 V_{DD} , TSRNG = 0
3.6 V	1.8 V

15.3 温度出力

温度回路の出力は、内蔵 A/D コンバータ (ADC) を使って計測します。ADC の入力には、温度回路からの出力用に 1 チャンネルが予約されています。詳細は [セクション 16.0 「A/D コンバータ \(ADC\) モジュール」](#) を参照してください。

15.4 ADC アクイジション時間

正確な温度を確実に計測するには、ADC 入力マルチプレクサが温度インジケータの出力に接続されてから 200 μ s 以上待って A/D 変換を開始する必要があります。また、温度インジケータ出力を連続して変換する場合、各変換動作の間に 200 μ s 待つ必要があります。

表 15-2: 温度インジケータ関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDFVR<1:0>		ADFVR<1:0>		173

凡例: 網掛けの部分は温度インジケータ モジュールでは使いません。

PIC16(L)F1764/5/8/9

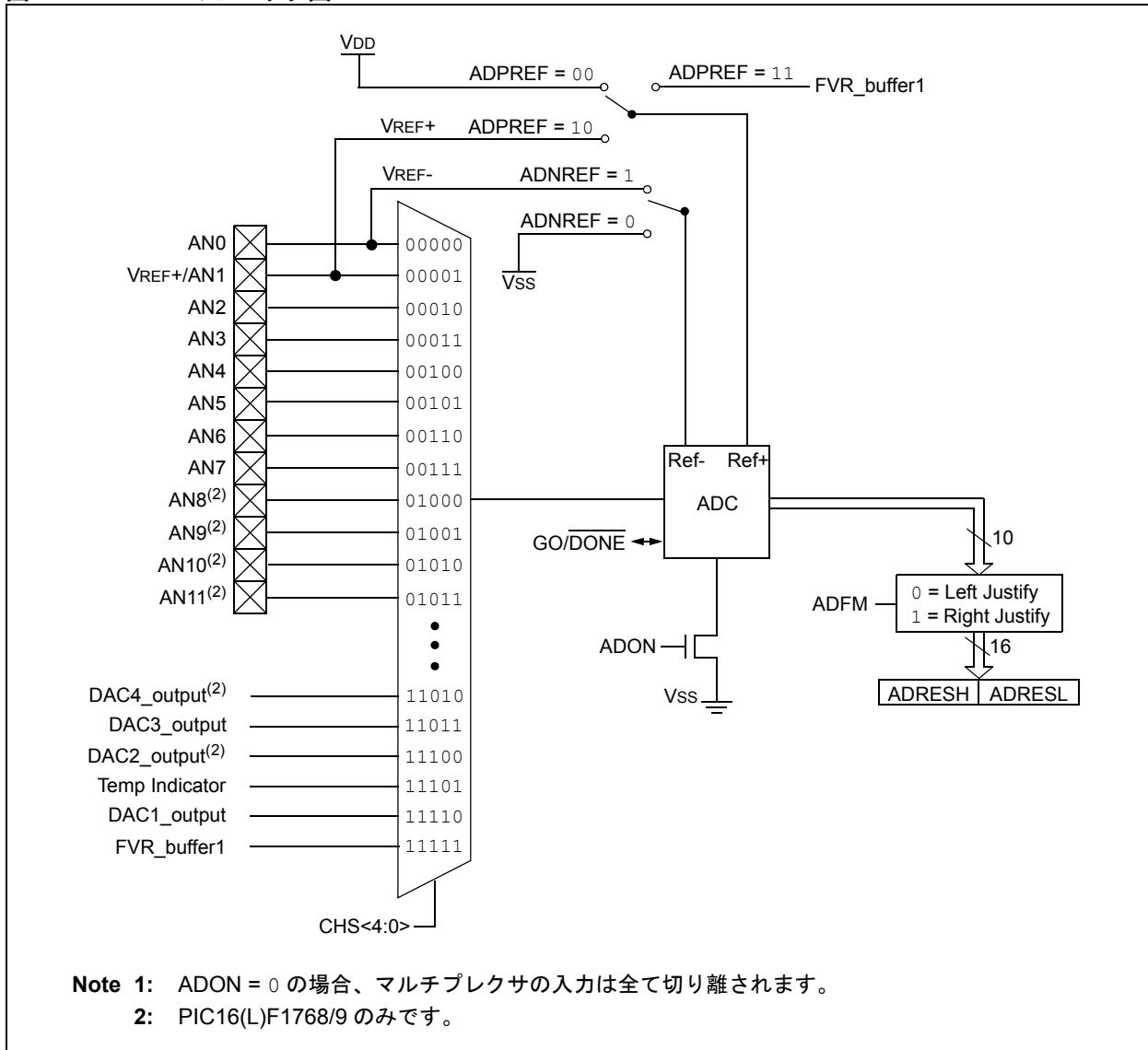
16.0 A/D コンバータ (ADC) モジュール

この ADC (A/D コンバータ) は、アナログ入力信号を 10 ビットのバイナリ値に変換します。本デバイスで使うアナログ入力は、多重化して 1 つのサンプルホールド回路に入力します。このサンプルホールド回路の出力を ADC の入力に接続します。ADC は逐次比較方式によって 10 ビットのバイナリ値を生成し、この変換結果を A/D 変換結果レジスタ (ADRESH:ADRESL レジスタペア) に格納します。図 16-1 に、ADC のブロック図を示します。

ADC の参照電圧はソフトウェアで選択可能であり、内部で生成するか外部から供給できます。

ADC は変換完了時に割り込みを生成できます。この割り込みを使って、デバイスをスリープから復帰させる事ができます。

図 16-1: ADC のブロック図



Note 1: ADON = 0 の場合、マルチプレクサの入力は全て切り離されます。
Note 2: PIC16(L)F1768/9 のみです。

16.1 ADC の設定

ADC の設定と使用に際しては、以下の機能を考慮する必要があります。

- ポートの設定
- チャンネルの選択
- ADC 参照電圧の選択
- ADC のクロック源
- 割り込みの制御
- 変換結果のフォーマット

16.1.1 ポートの設定

ADC は、アナログ信号とデジタル信号の両方の変換に使えます。アナログ信号を変換する場合、対応する TRIS ビットと ANSEL ビットをセットして I/O ピンをアナログに設定する必要があります。詳細は[セクション 11.0 「I/O ポート」](#)を参照してください。

Note: デジタル入力として定義されたピンにアナログ電圧を印加すると、入力バッファに過大な電流が流れる事があります。

16.1.2 チャンネルの選択

以下の最大 18 チャンネルから選択できます。

- AN<7:0> ピン
- AN<11:8> ピン (PIC16(L)F1768/9 のみ)
- 温度インジケータ
- DAC1_output と DAC3_output
- DAC2_output と DAC4_output (PIC16(L)F1768/9 のみ)
- FVR_buffer1

サンプルホールド回路にどのチャンネルを接続するかは、ADCON0 レジスタ ([レジスタ 16-1](#)) の CHS ビットで設定します。

チャンネルを変更した場合、次の変換を開始するまでに遅延が必要です。詳細は[セクション 16.2 「ADC の動作」](#)を参照してください。

16.1.3 ADC 正側参照電圧

ADCON1 レジスタの ADPREF ビットで正側参照電圧を制御します。以下の正側参照電圧が使えます。

- VREF+ ピン
- VDD
- FVR 2.048 V
- FVR 4.096 V (LF デバイスでは使えません)
- Vss

固定参照電圧の詳細は、[セクション 16.0 「A/D コンバータ \(ADC\) モジュール」](#)を参照してください。

16.1.4 ADC 負側参照電圧

ADCON1 レジスタの ADNREF ビットで負側参照電圧を設定します。以下の負側参照電圧を使用できます。

- VREF- ピン
- Vss

16.1.5 変換クロック

変換のクロック源は、ADCON1 レジスタの ADCS ビットで選択できます。クロックは以下の 7 種類から選択できます。

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC (内部 RC オシレータ)

1 ビットの変換に必要な時間を TAD と定義します。[図 16-2](#)に示すように、10 ビット全体の変換には 11.5 TAD が必要です。

正しく変換するには、TAD の仕様を適切に満たす必要があります。詳細は[表 36-16: A/D 変換の要件](#)を参照してください。[表 16-1](#)に、適切な ADC クロックの選択例を示します。

Note: FRC を使う場合を除き、システムクロック周波数を変更すると ADC クロック周波数も変化するため、A/D 変換結果に悪影響を与える可能性があります。

PIC16(L)F1764/5/8/9

表 16-1: ADC クロック周期 (TAD) に対するデバイスの動作周波数

ADC クロック周期 (TAD)		デバイスの動作周波数 (Fosc)					
ADC クロック源	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5 ns ⁽²⁾	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs
Fosc/4	100	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	4.0 μs
Fosc/8	001	0.5 μs ⁽²⁾	400 ns ⁽²⁾	0.5 μs ⁽²⁾	1.0 μs	2.0 μs	8.0 μs ⁽³⁾
Fosc/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs ⁽³⁾
Fosc/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs ⁽³⁾	32.0 μs ⁽²⁾
Fosc/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs ⁽³⁾	16.0 μs ⁽²⁾	64.0 μs ⁽²⁾
FRC	×11	1.0 ~ 6.0 μs ^(1, 4)	1.0 ~ 6.0 μs ^(1, 4)	1.0 ~ 6.0 μs ^(1, 4)	1.0 ~ 6.0 μs ^(1, 4)	1.0 ~ 6.0 μs ^(1, 4)	1.0 ~ 6.0 μs ^(1, 4)

凡例: 網掛けの値は推奨レンジ外です。

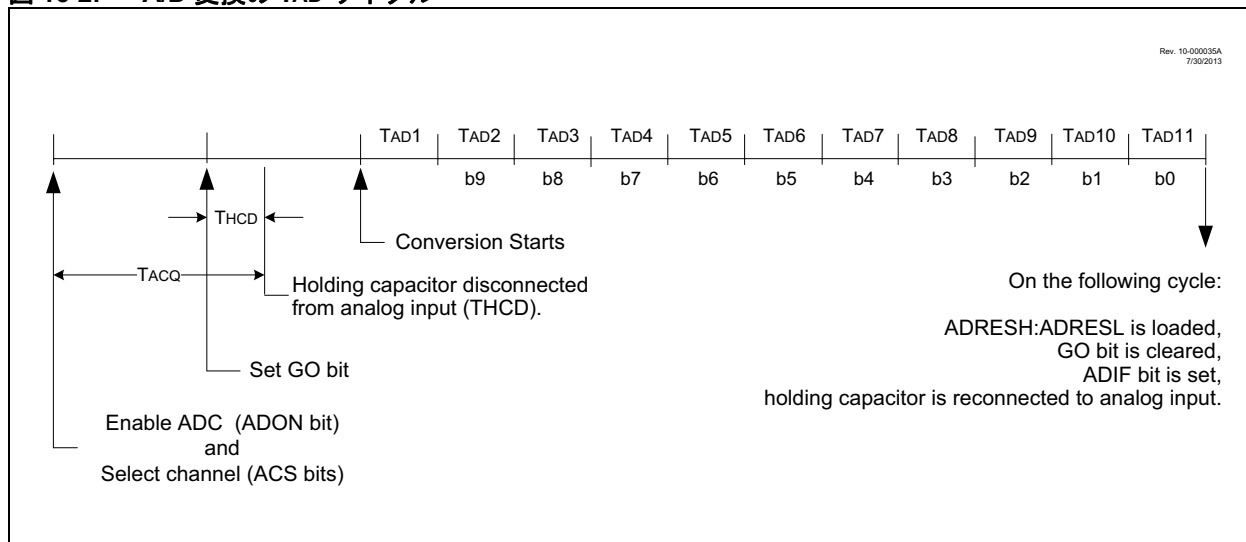
Note 1: FRC 入力源の代表的な TAD 値は、TAD パラメータを参照してください。

2: これらの値は TAD 時間の要件を満たしていません。

3: 推奨 TAD 時間の範囲外です。

4: システムクロック Fosc を ADC クロック源とすると、ADC クロックの周期 (TAD) と A/D 変換にかかる時間全体を最も短くできます。しかし、スリープ中も変換を実行する場合は FRC オシレータ信号源を使う必要があります。

図 16-2: A/D 変換の TAD サイクル



16.1.6 割り込み

ADC モジュールは、A/D 変換完了時に割り込みを生成できます。ADC 割り込みフラグは、PIR1 レジスタの ADIF ビットです。ADC 割り込みイネーブルは、PIE1 レジスタの ADIE ビットです。ADIF ビットはソフトウェアでクリアする必要があります。

- Note 1:** ADC 割り込みの有効 / 無効の設定にかかわらず、変換完了時には必ず ADIF ビットがセットされます。
- 2:** FRC オシレータが選択されている場合のみ、ADC はデバイスがスリープ中でも動作します。

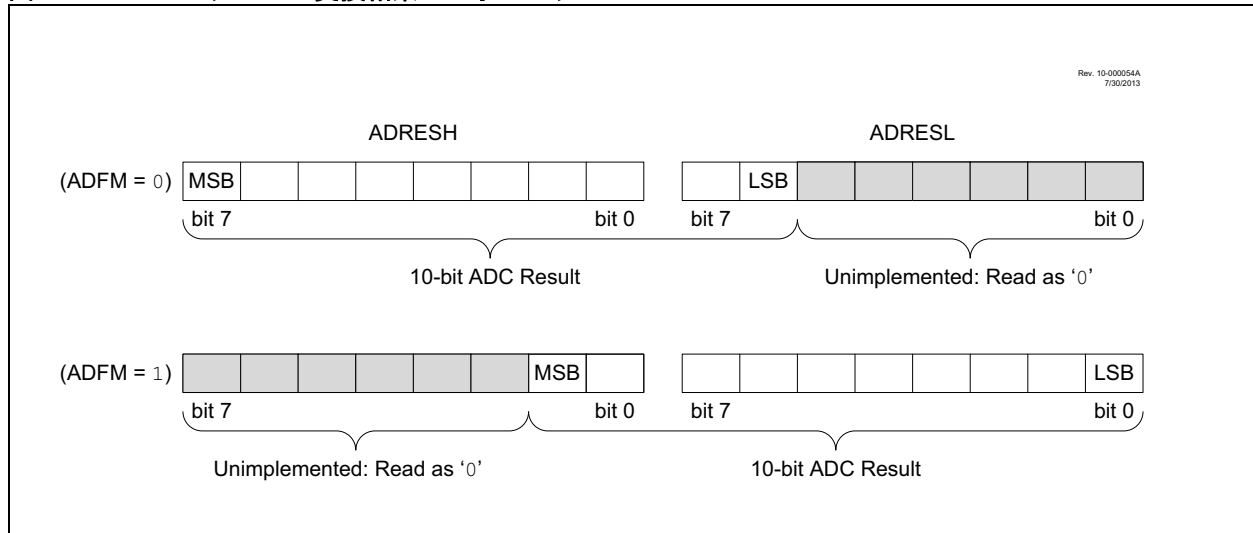
この割り込みは、デバイスの通常動作中またはスリープ中のどちらでも生成できます。デバイスがスリープ中に割り込みが発生すると、デバイスが復帰します。スリープから復帰すると、SLEEP 命令の次の命令が必ず実行されます。スリープからの復帰後にインラインコード実行を再開するには、PIE1 レジスタの ADIE ビットと INTCON レジスタの PEIE ビットをセットし、INTCON レジスタの GIE ビットをクリアしておく必要があります。これら 3 つのビットを全てセットしていると、実行は割り込みサービスルーチンに切り換わります。

16.1.7 変換結果のフォーマット

10 ビット A/D 変換の結果は、左詰めまたは右詰めどちらかのフォーマットで出力できます。出力フォーマットは、ADCON1 レジスタの ADFM ビットで設定します。

図 16-3 に、2 つの出力フォーマットを示します。

図 16-3: 10 ビット A/D 変換結果のフォーマット



PIC16(L)F1764/5/8/9

16.2 ADC の動作

16.2.1 変換の開始

ADC モジュールを有効にするには、ADCON0 レジスタの ADON ビットを「1」にセットする必要があります。ADCON0 レジスタの GO/DONE ビットを「1」にセットすると A/D 変換を開始します。

Note: ADC を ON にする命令と同じ命令で GO/DONE ビットをセットしてはいけません。セクション 16.2.6 「A/D 変換手順」を参照してください。

16.2.2 変換の完了

変換が完了すると、ADC モジュールは以下の動作を行います。

- GO/DONE ビットをクリアする
- ADIF 割り込みフラグビットをセットする
- ADRESH と ADRESL レジスタに変換結果を書き込む

16.2.3 変換の中止

変換を完了前に中止する場合、GO/DONE ビットをソフトウェアでクリアします。ADRESH および ADRESL レジスタは、途中まで完了した A/D 変換サンプルで更新されます。未完了のビットは、最後に変換したビットと同じ値です。

Note: デバイスリセットが発生すると、全てのレジスタはリセット状態に戻ります。このため、ADC モジュールは OFF になり、保留中の変換は全て中止されます。

16.2.4 スリープ中の ADC の動作

ADC モジュールはスリープ中も動作できます。ただし、ADC クロック源を FRC に設定する必要があります。FRC オシレータ信号源を選択した場合、ADC が変換を開始するまでの待ち時間は通常より 1 命令サイクル分長くなります。これによって SLEEP 命令の実行が可能となり、変換時のシステムノイズが軽減できます。ADC 割り込みが有効の場合、A/D 変換が完了するとデバイスがスリープから復帰します。ADC 割り込みが無効の場合、変換が完了すると ADON ビットはセットされたまま ADC モジュールが OFF になります。ADC クロック源が FRC 以外の場合、SLEEP 命令を実行すると現在の変換は中止され、ADON ビットがセットされたまま ADC モジュールが OFF になります。

16.2.5 自動変換トリガ

自動変換トリガを使うと、ソフトウェアを使わなくても定期的な ADC 計測を実行できます。選択したトリガ源に立ち上がりエッジが発生すると、ハードウェアが GO/DONE ビットをセットします。

自動変換トリガ源は、ADCON2 レジスタの TRIGSEL<4:0> ビットで選択します。

自動変換トリガを使っても、ADC の適切なタイミングが保証される訳ではありません。ADC のタイミング要件が満たされているかどうかは、ユーザが確認する必要があります。

表 16-2 に自動変換トリガ源を示します。

表 16-2: 自動変換トリガ源

トリガ源モジュール	信号名
CCP1	CCP1_trigger
CCP2 ⁽¹⁾	CCP2_trigger
Timer0	T0_overflow
Timer1	T1_overflow
Timer3	T3_overflow
Timer5	T5_overflow
Timer2	T2_postscaled
Timer4	T4_postscaled
Timer6	T6_postscaled
コンパレータ C1	sync_C1OUT
コンパレータ C2	sync_C2OUT
コンパレータ C3 ⁽¹⁾	sync_C3OUT
コンパレータ C4 ⁽¹⁾	sync_C4OUT
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
PWM3	PWM3OUT
PWM4 ⁽¹⁾	PWM4OUT
PWM5	PR/PH/OF/DC5_match
PWM6 ⁽¹⁾	PR/PH/OF/DC6_match

Note 1: PIC16(L)F1768/9 のみです。

16.2.6 A/D 変換手順

以下に ADC による A/D 変換の手順の例を示します。

1. ポートを設定する：
 - ピン出力ドライバを無効にする (TRIS レジスタ参照)。
 - ピンをアナログとして設定する (ANSEL レジスタ参照)。
2. ADC モジュールを設定する：
 - A/D 変換クロックを選択する。
 - 参照電圧を設定する。
 - ADC 入力チャンネルを選択する。
 - ADC モジュールを ON にする。
3. ADC 割り込みを設定する (任意):
 - ADC 割り込みフラグをクリアする。
 - ADC 割り込みを有効にする。
 - 周辺モジュール割り込みを有効にする。
 - グローバル割り込みを有効にする⁽¹⁾。
4. 必要なアキュイジション時間が経過するまで待つ⁽²⁾。
5. GO/DONE ビットをセットして変換を開始する。
6. 以下のどちらかの方法で、A/D 変換の完了を待つ：
 - GO/DONE ビットをポーリングする。
 - ADC 割り込みを待つ (割り込みが有効の場合)。
7. A/D 変換の結果を読み出す。
8. ADC 割り込みフラグをクリアする (割り込みが有効の場合は必須)。

例 16-1: A/D 変換

```

;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL    ADCON1        ;
MOVLW     B'11110000'    ;Right justify, FRC
                                ;oscillator

MOVWF     ADCON1        ;Vdd and Vss Vref
BANKSEL    TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL    ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL    ADCON0      ;
MOVLW     B'00000001'   ;Select channel AN0
MOVWF     ADCON0      ;Turn ADC On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,ADGO  ;Start conversion
BTFSC    ADCON0,ADGO  ;Is conversion done?
GOTO     $-1          ;No, test again
BANKSEL    ADRESH      ;
MOVF     ADRESH,W     ;Read upper 2 bits
MOVWF    RESULTHI     ;store in GPR space
BANKSEL    ADRESL      ;
MOVF     ADRESL,W     ;Read lower 8 bits
MOVWF    RESULTLO     ;Store in GPR space
    
```

Note 1: スリープからの復帰後にインラインコード実行を再開させる場合、グローバル割り込みは無効でもかまいません。

2: [セクション 16.4 「ADC アキュイジションの要件」](#) を参照してください。

PIC16(L)F1764/5/8/9

16.3 レジスタ定義 : ADC 制御

レジスタ 16-1: ADCON0: ADC 制御レジスタ 0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7 **未実装:** 「0」として読み出し

bit 6-2 **CHS<4:0>:** アナログ チャンネル選択ビット

11111 = FVR (固定参照電圧) バッファ 1 出力⁽²⁾

11110 = DAC1_output⁽¹⁾

11101 = 温度インジケータ⁽³⁾

11100 = DAC2_output^(1, 5)

11011 = DAC3_output⁽⁴⁾

11010 = DAC4_output^(4, 5)

11001 = 予約済みチャンネル接続なし

•

•

•

01100 = 予約済みチャンネル接続なし

01011 = AN11⁽⁵⁾

01010 = AN10⁽⁵⁾

01001 = AN9⁽⁵⁾

01000 = AN8⁽⁵⁾

00111 = AN7

00110 = AN6

00101 = AN5

00100 = AN4

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

bit 1 **GO/DONE:** A/D 変換ステータスビット

1 = A/D 変換サイクルを実行中である。このビットをセットすると A/D 変換サイクルが開始する

このビットは、A/D 変換が完了するとハードウェアによって自動的にクリアされます。

0 = A/D 変換サイクルが完了している (進行中でない)

bit 0 **ADON:** ADC イネーブルビット

1 = ADC を有効にする

0 = ADC を無効にし、消費電流をゼロにする

Note 1: 詳細は[セクション 17.0 「5 ビット D/A コンバータ \(DAC\) モジュール」](#)を参照してください。

2: 詳細は[セクション 14.0 「固定参照電圧 \(FVR\)」](#)を参照してください。

3: 詳細は[セクション 15.0 「温度インジケータ モジュール」](#)を参照してください。

4: 詳細は[セクション 18.0 「10 ビット D/A コンバータ \(DAC\) モジュール」](#)を参照してください。

5: PIC16(L)F1768/9 のみです。

レジスタ 16-2: ADCON1: ADC 制御レジスタ 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>		—	ADNREF	ADPREF<1:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7 **ADFM:** A/D 変換結果フォーマット選択ビット
 1 = 右詰め、変換結果を書き込むと ADRESH の上位 6 ビットが「0」に設定される
 0 = 左詰め、変換結果を書き込むと ADRESL の下位 6 ビットが「0」に設定される
- bit 6-4 **ADCS<2:0>:** ADC 変換クロック選択ビット
 111 = FRC(内部の RC オシレータから供給されるクロック)
 110 = Fosc/64
 101 = Fosc/16
 100 = Fosc/4
 011 = FRC(内部の RC オシレータから供給されるクロック)
 010 = Fosc/32
 001 = Fosc/8
 000 = Fosc/2
- bit 3 **未実装:** 「0」として読み出し
- bit 2 **ADNREF:** ADC 負側参照電圧コンフィグレーション ビット
 1 = VREF- を外部 VREF- ピンに接続する
 0 = VREF- を VSS に接続する
- bit 1-0 **ADPREF<1:0>:** ADC 正側参照電圧コンフィグレーション ビット
 11 = VREF+ を内部の固定参照電圧 (FVR) モジュールに接続する ⁽¹⁾
 10 = VREF+ を外部 VREF+ ピンに接続する ⁽¹⁾
 01 = 予約済み
 00 = VREF+ を VDD に接続する

Note 1: 正側参照電圧の入力源として VREF+ ピンを選択する場合、最小電圧値に注意してください。詳細は表 36-16: A/D 変換の要件を参照してください。

PIC16(L)F1764/5/8/9

レジスタ 16-3: **ADCON2: ADC 制御レジスタ 2**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0
TRIGSEL<4:0> ⁽¹⁾					—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-3 **TRIGSEL<4:0>: 自動変換トリガ選択ビット⁽¹⁾**

```

11111 = 予約済み
.
.
.
11011 = 予約済み
11010 = PWM6 – OF6_match(2)
11001 = PWM6 – PH6_match(2)
11000 = PWM6 – PR6_match(2)
10111 = PWM6 – DC6_match(2)
10110 = PWM5 – OF5_match
10101 = PWM5 – PH5_match
10100 = PWM5 – PR5_match
10011 = PWM5 – DC5_match
10010 = PWM4 – PWM4OUT(2)
10001 = PWM3 – PWM3OUT
10000 = CCP2 – CCP2_trigger(2)
01111 = CCP1 – CCP1_trigger
01110 = CLC3 – LC3_out
01101 = CLC2 – LC2_out
01100 = CLC1 – LC1_out
01011 = コンパレータ C4 – sync_C4OUT(2)
01010 = コンパレータ C3 – sync_C3OUT(2)
01001 = コンパレータ C2 – sync_C2OUT
01000 = コンパレータ C1 – sync_C1OUT
00111 = Timer6 – T6_postscaled
00110 = Timer5 – T5_overflow
00101 = Timer4 – T4_postscaled
00100 = Timer3 – T3_overflow
00011 = Timer2 – T2_postscaled
00010 = Timer1 – T1_overflow
00001 = Timer0 – T0_overflow
00000 = 自動変換トリガを選択しない
  
```

bit 3-0 **未実装:** 「0」として読み出し

- Note 1:** どのトリガ源も立ち上がりエッジを検出します。
2: PIC16(L)F1768/9 のみです。他の製品では予約済みです。

PIC16(L)F1764/5/8/9

レジスタ 16-4: ADRESH: A/D 変換結果レジスタの上位 (High) (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **ADRES<9:2>**: A/D 変換結果レジスタビット
10 ビット変換結果の上位 8 ビット

レジスタ 16-5: ADRESL: A/D 変換結果レジスタの下位 (Low) (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **ADRES<1:0>**: A/D 変換結果レジスタビット
10 ビット変換結果の下位 2 ビット

bit 5-0 **予約済み**: 使用不可

PIC16(L)F1764/5/8/9

レジスタ 16-6: ADRESH: A/D 変換結果レジスタの上位 (High) (ADRESH) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-2 **予約済み** : 使用不可
bit 1-0 **ADRES<9:8>**: A/D 変換結果レジスタビット
10 ビット変換結果の上位 2 ビット

レジスタ 16-7: ADRESL: A/D 変換結果レジスタの下位 (Low) (ADRESL) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **ADRES<7:0>**: A/D 変換結果レジスタビット
10 ビット変換結果の下位 8 ビット

16.4 ADC アクイジションの要件

ADC が仕様上の精度を実現するには、電荷ホールドコンデンサ (CHOLD) を入力チャンネルの電圧レベルまで完全に充電する必要があります。図 16-4 に、アナログ入力モデルを示します。ソース インピーダンス (R_s) と内部サンプリングスイッチのインピーダンス (R_{SS}) は、このコンデンサ (CHOLD) の充電時間に直接影響します。サンプリングスイッチのインピーダンス (R_{SS}) は、デバイス電圧 (V_{DD}) によって変化します (図 16-4 参照)。アナログソースの推奨最大インピーダンスは 10 k Ω です。ソース インピーダンスが低下する

と、アクイジション時間が短くなる場合があります。アナログ入力チャンネルを選択 (または変更) した場合、変換開始前に ADC アクイジションを実行する必要があります。最小アクイジション時間は、式 16-1 で求めます。この式では、1/2 LSb 誤差の適用を前提としています (ADC は 1,024 ステップ)。1/2 LSb の誤差は、ADC で仕様上の分解能を実現できる最大許容誤差です。

式 16-1: アクイジション時間の例

仮定: $Temperature = 50^{\circ}C$ and external impedance of $10k\Omega$ $5.0V V_{DD}$

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(Temperature - 25^{\circ}C)(0.05\mu s/^{\circ}C)] \end{aligned}$$

TC の値は下式から近似的に求められる:

$$V_{APPLIED} \left(1 - \frac{1}{(2^n + 1) - 1} \right) = V_{CHOLD} \quad ;[1] V_{CHOLD} \text{ を } 1/2 \text{ lsb 以内まで充電}$$

$$V_{APPLIED} \left(1 - e^{-\frac{TC}{RC}} \right) = V_{CHOLD} \quad ;[2] V_{APPLIED} \text{ に対する } V_{CHOLD} \text{ の充電の応答}$$

$$V_{APPLIED} \left(1 - e^{-\frac{TC}{RC}} \right) = V_{APPLIED} \left(1 - \frac{1}{(2^n + 1) - 1} \right) \quad ;[1] \text{ と } [2] \text{ を組み合わせ}$$

Note: $n = \text{ADC のビット数}$

TC を求めると:

$$\begin{aligned} T_C &= -CHOLD(R_{IC} + R_{SS} + R_s) \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

従って:

$$\begin{aligned} T_{ACQ} &= 2\mu s + 892ns + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)] \\ &= 4.62\mu s \end{aligned}$$

Note 1: 参照電圧 (V_{REF}) は自己相殺されるため、上式には影響しません。

2: 電荷ホールドコンデンサ (CHOLD) は、変換が完了するたびに放電される訳ではありません。

3: アナログソースの推奨最大インピーダンスは 10 k Ω です。この値はピンのリーク電流仕様を満たすために必要です。

PIC16(L)F1764/5/8/9

図 16-4: アナログ入力モデル

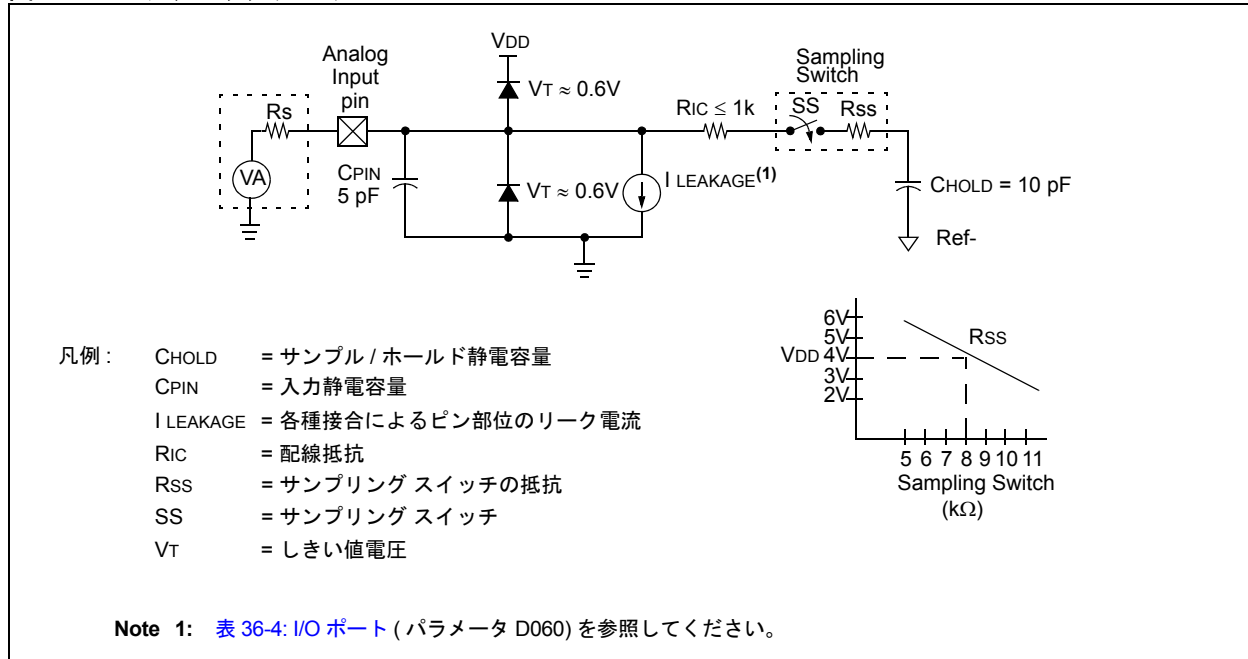


図 16-5: ADC の伝達関数

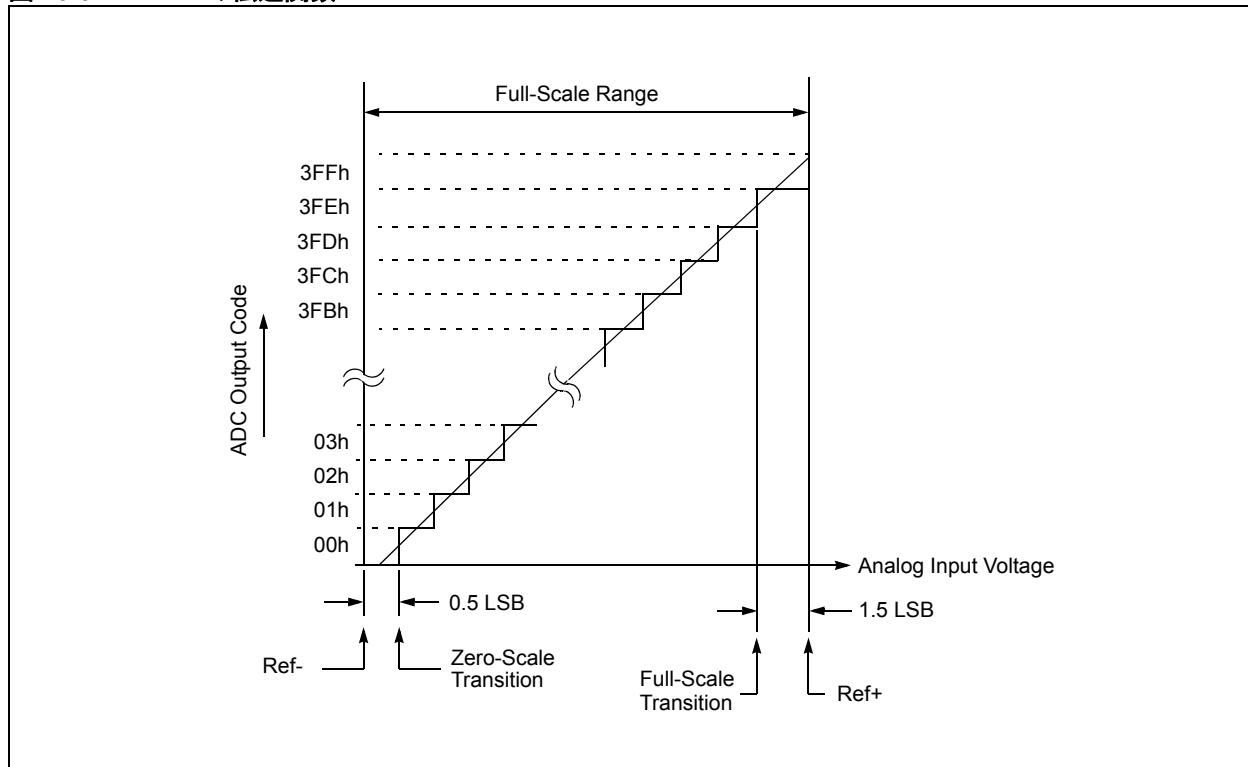


表 16-3: ADC 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 容記載 ページ
ADCON0	—	CHS<4:0>					GO/DONE	ADON	182
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		183
ADCON2	TRIGSEL<4:0>					—	—	—	184
ADRESH	A/D 変換結果レジスタ上位								185、186
ADRESL	A/D 変換結果レジスタ下位								185、186
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173
DAC1CON0	EN	FM	OE1	—	PSS<1:0>		NSS<1:0>		193

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による。網掛けの部分は ADC モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

17.0 5ビット D/A コンバータ (DAC) モジュール

この D/A コンバータ (DAC) は、選択可能な 32 出力レベルに応じて、入力源に対して比例した可変参照電圧を供給します。

DAC の入力には、以下のものを接続できます。

- 外部 VREF ピン
- VDD 電源電圧
- FVR(固定参照電圧)

DAC の出力は、以下のいずれかで参照電圧として使うように設定できます。

- コンパレータ正入力
- オペアンプの反転および非反転入力
- ADC 入力チャンネル
- DACxOUT1 ピン

表 17-1: 利用可能な 5 ビット DAC

デバイス	D3	D4
PIC16(L)F1764	●	
PIC16(L)F1765	●	
PIC16(L)F1768	●	●
PIC16(L)F1769	●	●

DAC を有効にするには、DACxCON0 レジスタの EN ビットをセットします。

17.1 出力電圧の選択

DAC には 32 レベルの電圧レンジがあります。これら 32 のレベルは、DACxREF レジスタの REF<4:0> ビットで設定します。

DAC の出力電圧は、式 17-1 で求めます。

式 17-1: DAC 出力電圧

DACxEN = 1 の場合

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACxR[4:0]}{2^5} \right) + V_{SOURCE-}$$

*V*_{SOURCE+} = *V*_{DD}、*V*_{REF}、*FVR BUFFER 2* のいずれか

*V*_{SOURCE-} = *V*_{SS}

17.2 レシオメトリック出力レベル

DAC の出力値は、抵抗ラダーによって生成します。このラダーの両端は正負の参照電圧入力源に接続されています。どちらか一方の入力源の電圧が変動すると、DAC 出力値にも同様の変動が生じます。

ラダーを構成する各抵抗の値は、表 36-20: 10 ビット D/A コンバータ (DAC) の仕様に記載しています。

17.3 DAC 参照電圧出力

DAC の電圧は、DACxCON0 レジスタの OE1 ビットをセットすると、DACxOUT1 ピンに出力できます。DAC 電圧を DACxOUT1 ピンに出力すると、そのピンのデジタル出力バッファおよびデジタル入力より自動的に優先されます。DAC 電圧出力に設定した DACxOUT1 ピンを読み出すと、常に「0」が返されます。

電流駆動能力が限られているため、DAC 電圧を DACxOUT1 ピンに出力して外部接続する場合はバッファを使う必要があります。図 17-2 に、バッファの使い方の例を示します。

図 17-1: DAC のブロック図

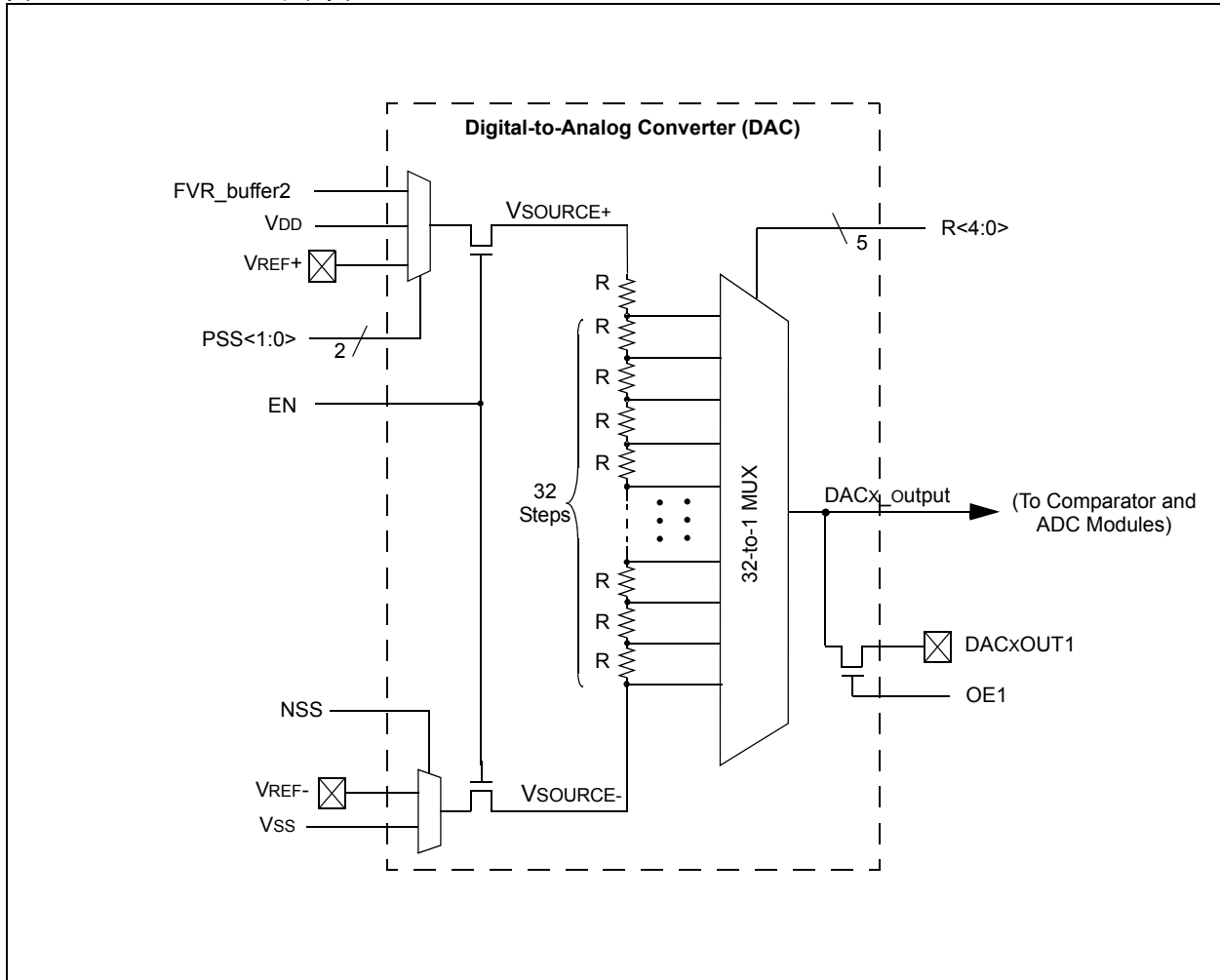
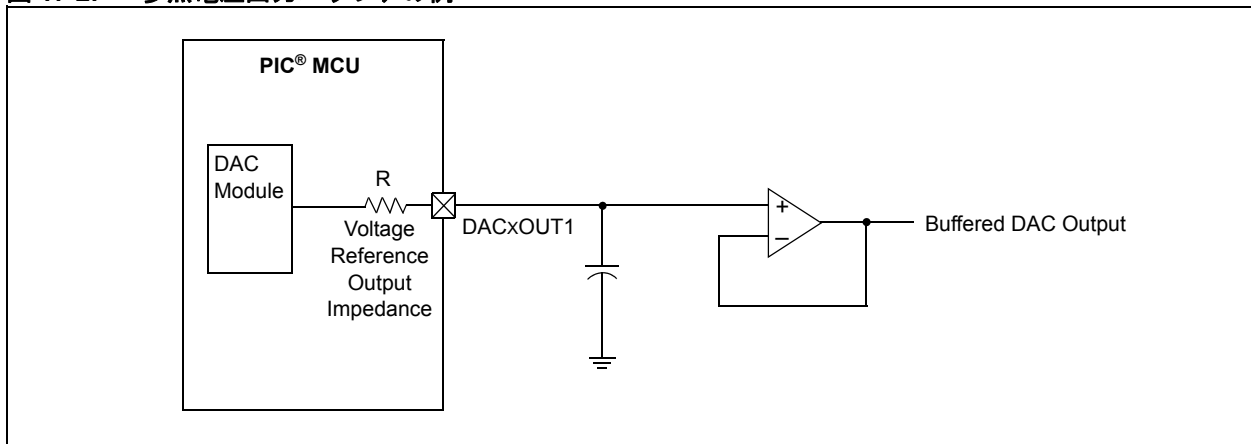


図 17-2: 参照電圧出力バッファの例



PIC16(L)F1764/5/8/9

17.4 スリープ中の動作

DAC はスリープ中も動作を継続します。割り込みまたはウォッチドッグ タイマのタイムアウトによってデバイスがスリープ状態から復帰する際、DACXCON0 レジスタの内容は変更されません。スリープ中の消費電流を最小にするには、参照電圧を無効にします。

17.5 リセットの影響

デバイスリセットの影響は以下の通りです。

- DAC を無効にする。
- DAC 出力電圧を DACXOUT1 ピンから切り離す。
- REF<4:0> 参照電圧制御ビットをクリアする。

17.6 レジスタ定義 :DAC 制御

表 17-2 に 5 ビット DAC 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1 「レジスタおよびビット命名法」](#) を参照してください。

表 17-2:

周辺モジュール	ビット名の接頭辞
DAC3	DAC3
DAC4 ⁽¹⁾	DAC4

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 17-1: DACxCON0: DACx 制御レジスタ 0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
EN	—	OE1	—	PSS<1:0>		—	NSS
bit 7							bit 0

凡例:

R = 読み出し可能ビット

W = 書き込み可能ビット

U = 未実装ビット、「0」として読み出し

u = ビットは不変

x = ビットは未知

-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット

「0」= ビットはクリア

- bit 7 **EN:** DAC イネーブルビット
 1 = DAC を有効にする
 0 = DAC を無効にする
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **OE1:** DAC 電圧出力イネーブルビット
 1 = DAC 電圧レベルを DACxOUT1 ピンにも出力する
 0 = DAC 電圧レベルを DACxOUT1 ピンから切り離す
- bit 4 **未実装:** 「0」として読み出し
- bit 3-2 **PSS<1:0>:** DAC 正参照電圧源選択ビット
 11 = 予約済み (使用禁止)
 10 = FVR バッファ 2 出力
 01 = VREF+ ピン
 00 = VDD
- bit 1 **未実装:** 「0」として読み出し
- bit 0 **NSS:** DAC 負参照電圧源選択ビット
 1 = VREF- ピン
 0 = VSS

レジスタ 17-2: DACxREF: DACx 参照電圧出力選択レジスタ

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	REF<4:0>				
bit 7							bit 0

凡例:

R = 読み出し可能ビット

W = 書き込み可能ビット

U = 未実装ビット、「0」として読み出し

u = ビットは不変

x = ビットは未知

-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット

「0」= ビットはクリア

- bit 7-5 **未実装:** 「0」として読み出し
- bit 4-0 **REF<4:0>:** DACx 参照電圧出力選択ビット (式 17-1 参照)

PIC16(L)F1764/5/8/9

表 17-3: DACx モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
DAC3CON0	EN	—	OE1	—	PSS<1:0>		—	NSS	193
DAC4CON0 ⁽¹⁾	EN	—	OE1	—	PSS<1:0>		—	NSS	193
DAC3REF	—	—	—	REF<4:0>					193
DAC4REF ⁽¹⁾	—	—	—	REF<4:0>					193

凡例: — = 未実装、「0」として読み出し。網掛けの部分は DACx モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

18.0 10ビットD/Aコンバータ(DAC)モジュール

10ビットD/Aコンバータ(DAC)は、選択可能な1024出力レベルに応じて、入力源に対して比例した可変参照電圧を供給します。

DACの入力には、以下のものを接続できます。

- 外部VREFピン
- VDD電源電圧
- FVR(固定参照電圧)

DACの出力は、以下のいずれかで参照電圧として使うように設定できます。

- コンパレータ正入力
- ADC入力チャンネル
- DACxOUT1ピン
- オペアンプ

D/Aコンバータを有効にするには、DACxCON0レジスタのENビットをセットします。

表 18-1: 利用可能な10ビットDAC

デバイス	D1	D2
PIC16(L)F1764	•	
PIC16(L)F1765	•	
PIC16(L)F1768	•	•
PIC16(L)F1769	•	•

18.1 出力電圧レベルの選択

このDACは1024の電圧レベルを持っています。これらの電圧レベルはDACxREFHおよびDACxREFLレジスタに格納される10ビットの参照選択ワードで設定します。この10ビットワードは左詰めでも右詰めでもかまいません。詳細は[セクション 18.4「DAC参照選択ワードの位置合わせ」](#)を参照してください。

DACの出力電圧は、[式 18-1](#)で求めます。

式 18-1: DAC出力電圧

$EN = 1$ の場合

$$DACx_output = \left((VSOURCE+ - VSOURCE-) \times \frac{DACxR[9:0]}{2^{10}} \right) + VSOURCE-$$

$VSOURCE+ = VDD, VREF+, FVR_buffer2$ のいずれか

$VSOURCE- = Vss$ または $VREF-$

18.2 参照電圧に比例する出力電圧

DACの出力電圧は抵抗ラダーによって生成します。このラダーの両端は正負の電圧源に接続されています。どちらか一方の入力源の電圧が変動すると、DAC出力値にも同様の変動が生じます。

ラダーを構成する各抵抗の値は、[表 36-20](#)に記載しています。

18.3 DAC出力

DAC電圧はこれを使う周辺モジュールから常時利用できます。DACの電圧は、DACxCON0レジスタのOE1ビットをセットすると、DACxOUT1ピンに出力できます。DAC電圧をDACxOUT1ピンに出力すると、そのピンのデジタル出力バッファおよびデジタル入力より自動的に優先されます。DAC電圧出力に設定したDACxOUT1ピンを読み出すと、常に「0」が返されます。

電流駆動能力が限られているため、DAC電圧をDACxOUT1ピンに出力して外部接続する場合はバッファを使う必要があります。[図 18-3](#)にバッファの使用例を示します。

18.4 DAC参照選択ワードの位置合わせ

DAC参照選択ワードは左詰め、右詰めのどちらにも設定できます。DACxCON0レジスタのFMビットをセットすると、上位8ビットがDACxREFHレジスタに、下位2ビットがDACxREFLレジスタにそれぞれ左詰めで格納され、10ビットワードは左詰めになります。DACxCON0レジスタのFMビットをクリアすると、下位8ビットがDACxREFLレジスタに、上位2ビットがDACxREFHレジスタにそれぞれ右詰めで格納され、10ビットワードは右詰めになります。[図 18-1](#)を参照してください。

DACxREFLレジスタとDACxREFHレジスタはダブルバッファ構成です。どちらのレジスタに書き込んでも即座には影響を与えません。DACLDレジスタのDACxLDビットに「1」を書き込むと、DACxREFLおよびDACxREFHレジスタの内容をバッファに転送し、それによって10ビットのDAC参照選択ワードを同時に書き換えます。

PIC16(L)F1764/5/8/9

図 18-1: DAC 参照選択ワードの位置合わせ

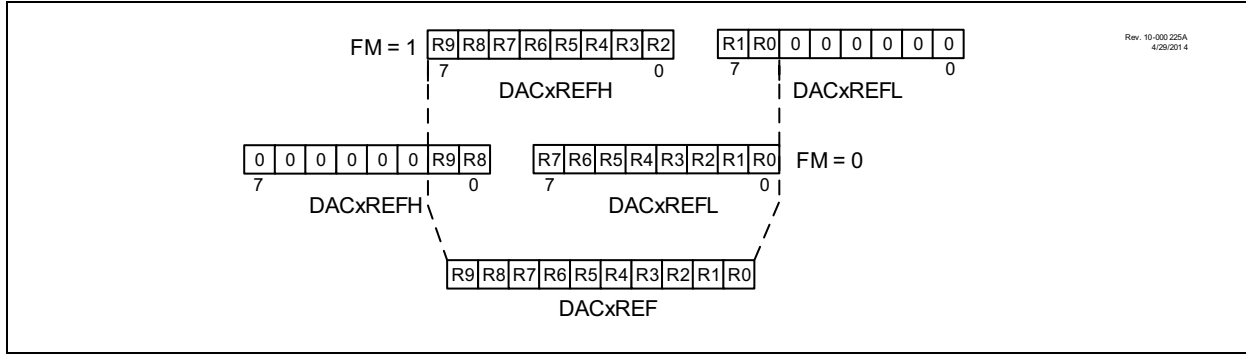


図 18-2: DAC のブロック図

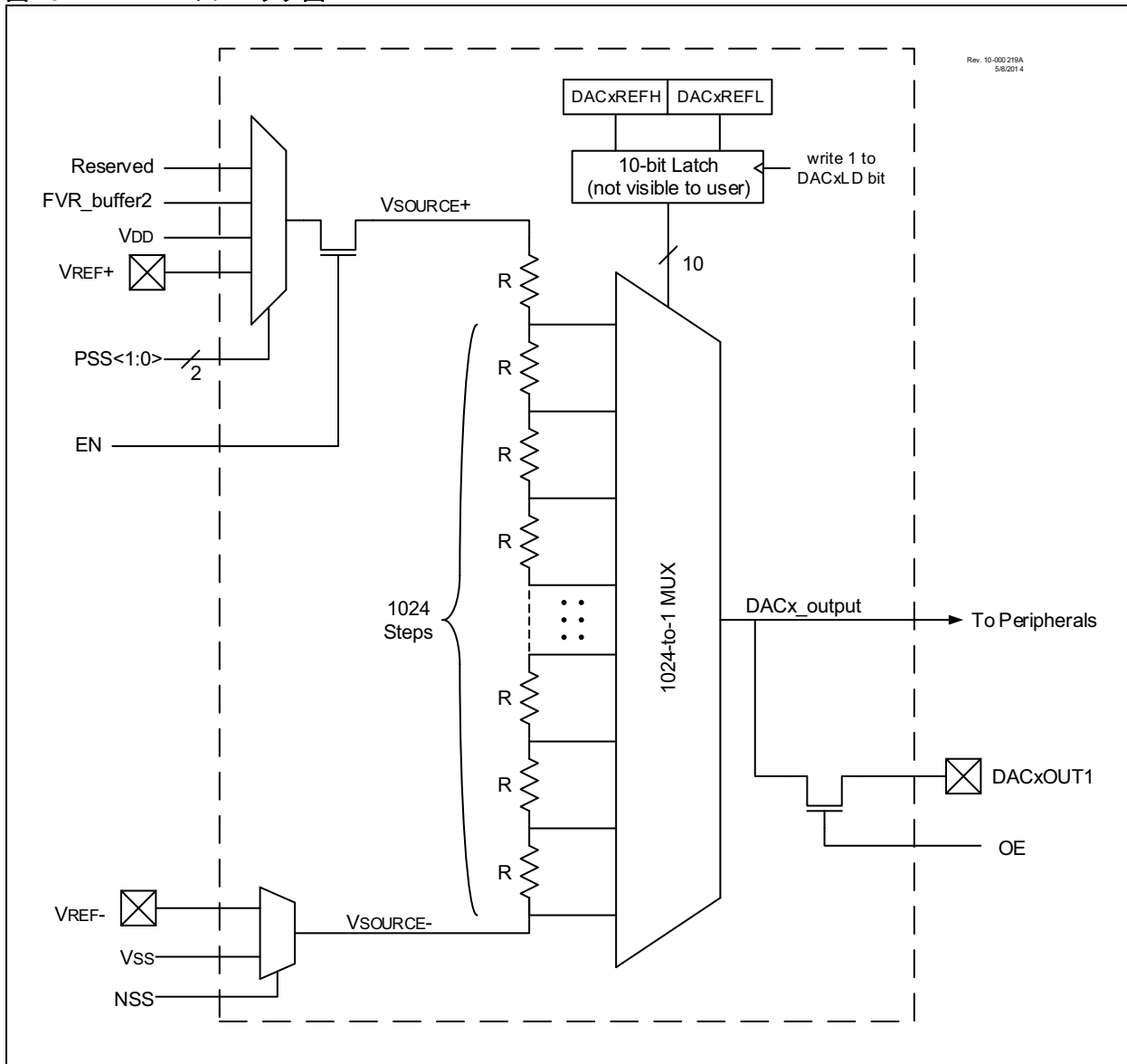
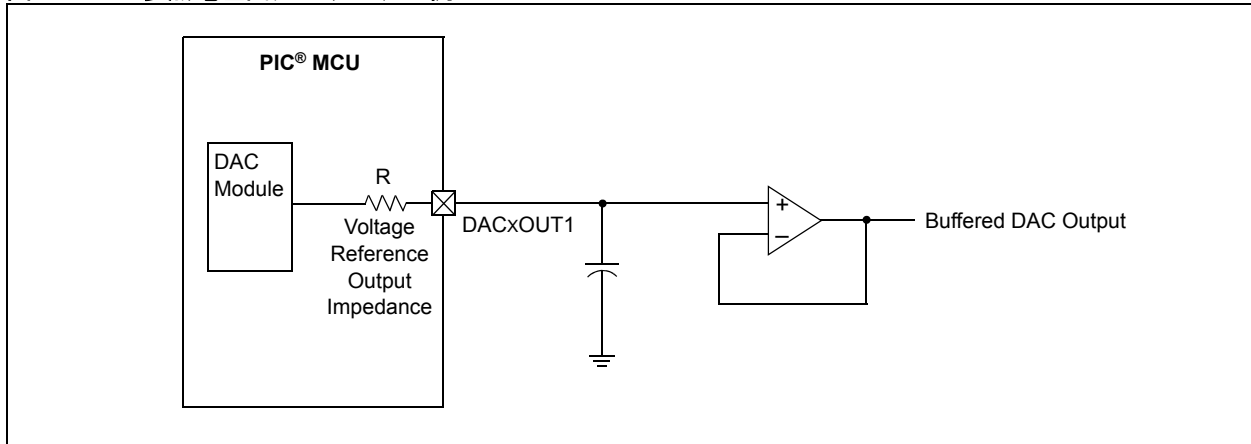


図 18-3: 参照電圧出力バッファの例



PIC16(L)F1764/5/8/9

18.5 スリープ中の動作

割り込みまたはウォッチドッグ タイマのタイムアウトによってデバイスがスリープ状態から復帰する際、DACxCON0 レジスタの内容は変更されません。スリープ中の消費電流を最小にするには、参照電圧を無効にします。

18.6 リセットの影響

デバイスリセットの影響は以下の通りです。

- DAC を無効にする。
- DAC 出力電圧を DACxOUT1 ピンから切り離す。
- REF<9:0> 参照電圧制御ビットをクリアする。

18.7 レジスタ定義 : DAC 制御

表 18-2 に 10 ビット DAC 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1 「レジスタおよびビット命名法」](#) を参照してください。

表 18-2:

周辺モジュール	ビット名の接頭辞
DAC1	DAC1
DAC2 ⁽¹⁾	DAC2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 18-1: DACxCON0: DAC 制御レジスタ 0

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
EN	FM	OE1	—	PSS<1:0>		—	NSS
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7 **EN:** DAC イネーブルビット
 1 = DACx を有効にする
 0 = DACx を無効にする

bit 6 **FM:** DAC 参照ワードフォーマット ビット
 1 = DACx 参照選択ワードを左詰めにする
 0 = DACx 参照選択ワードを右詰めにする

bit 5 **OE1:** DAC 電圧出力イネーブルビット
 1 = DACx 電圧レベルを DACxOUT1 ピンにも出力する
 0 = DACx 電圧レベルを DACxOUT1 ピンから切り離す

bit 4 **未実装:** 「0」として読み出し

bit 3-2 **PSS<1:0>:** DAC 正参照電圧源選択ビット
 11 = 予約済み使用不可
 10 = FVR_buffer2
 01 = VREF+ ピン
 00 = VDD

bit 1 **未実装:** 「0」として読み出し

bit 0 **NSS:** DAC 負の参照電圧源選択ビット
 1 = VREF- ピン
 0 = VSS

PIC16(L)F1764/5/8/9

レジスタ 18-2: DACxREFH: DAC 参照電圧選択上位レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
REF<9:x> (x は FM ビットによって異なる)							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

FM = 1(左詰め)の場合

bit 7-0 **REF<9:2>**: DAC 参照電圧出力選択ビット
DACxOUT1 = f(REF<9:0>) (式 18-1 参照)

FM = 0(右詰め)の場合

bit 7-2 **未実装**: 「0」として読み出し
bit 1-0 **REF<9:8>**: DAC 参照電圧出力選択ビット
DACxOUT1 = f(REF<9:0>) (式 18-1 参照)

レジスタ 18-3: DACxREFL: DAC 参照電圧選択下位レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
REF<x-1:0> (x は FM ビットによって異なる)							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

FM = 1(左詰め)の場合

bit 7-6 **REF<1:0>**: DAC 参照電圧出力選択ビット
DACxOUT1 = f(REF<9:0>) (式 18-1 参照)

bit 5-0 **未実装**: 「0」として読み出し

FM = 0(右詰め)の場合

bit 7-0 **REF<7:0>**: DAC 参照電圧出力選択ビット
DACxOUT1 = f(REF<9:0>) (式 18-1 参照)

レジスタ 18-4: DACLD: DAC バッファ書き込みレジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	DAC2LD ⁽¹⁾	DAC1LD
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

bit 7-2 **未実装:** 「0」として読み出し

bit 1 **DAC2LD:** DAC2 ダブルバッファ書き込みビット⁽¹⁾

1 = DAC2REFHL: DAC2REFL の値をダブルバッファに転送する。ビットはハードウェアによって自動的にクリアされる

0 = DAC2REFHL: DAC2REFL のダブルバッファを変更せずに維持する

bit 0 **DAC1LD:** DAC1 ダブルバッファ書き込みビット

1 = DAC1REFHL: DAC1REFL の値をダブルバッファに転送する。ビットはハードウェアによって自動的にクリアされる

0 = DAC1REFHL: DAC1REFL のダブルバッファを変更せずに維持する

Note 1: PIC16(L)F1768/9 のみです。

表 18-3: DACx モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
DAC1CON0	EN	FM	OE1	—	PSS<1:0>	—	—	NSS	199
DAC2CON0 ⁽¹⁾	EN	FM	OE1	—	PSS<1:0>	—	—	NSS	199
DAC1REFH	REF<9:x> (x は FM ビットによって異なる)								200
DAC2REFH ⁽¹⁾	REF<9:x> (x は FM ビットによって異なる)								200
DAC1REFL	REF<x-1:0> (x は FM ビットによって異なる)								200
DAC2REFL ⁽¹⁾	REF<x-1:0> (x は FM ビットによって異なる)								200
DACLD	—	—	—	—	—	—	DAC2LD ⁽¹⁾	DAC1LD	201

凡例: — = 未実装、「0」として読み出し。網掛けの部分は DACx モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

19.0 コンパレータ モジュール

コンパレータは、2つのアナログ電圧を比較してその相対的な大きさをデジタル値で表現する事によって、各種アナログ回路をデジタル回路に接続する役割を果たします。コンパレータはプログラム実行から独立したアナログ機能を提供するため、非常に便利なミクストシグナル構成要素です。アナログ コンパレータ モジュールには、以下の特長があります。

- 独立したコンパレータ制御
- プログラマブルな入力選択
- コンパレータ出力を内部 / 外部で利用可能
- プログラマブルな出力極性
- 状態変化割り込み
- スリープからの復帰
- 速度 / 電力をプログラミングにより最適化可能
- PWM シャットダウン
- プログラマブルおよび固定の参照電圧

19.1 コンパレータの概要

図 19-1 に、シングル コンパレータの構成およびアナログ入力レベルとデジタル出力の関係を示します。 V_{IN+} のアナログ電圧が V_{IN-} のアナログ電圧よりも低い場合、コンパレータ出力はデジタル Low レベルです。 V_{IN+} のアナログ電圧が V_{IN-} のアナログ電圧よりも高い場合、コンパレータ出力はデジタル High レベルです。

表 19-1 に、このデバイスで利用可能なコンパレータを示します。

表 19-1: 利用可能なコンパレータ

デバイス	C1	C2	C3	C4
PIC16(L)F1764	●	●		
PIC16(L)F1765	●	●		
PIC16(L)F1768	●	●	●	●
PIC16(L)F1769	●	●	●	●

図 19-1: シングル コンパレータ

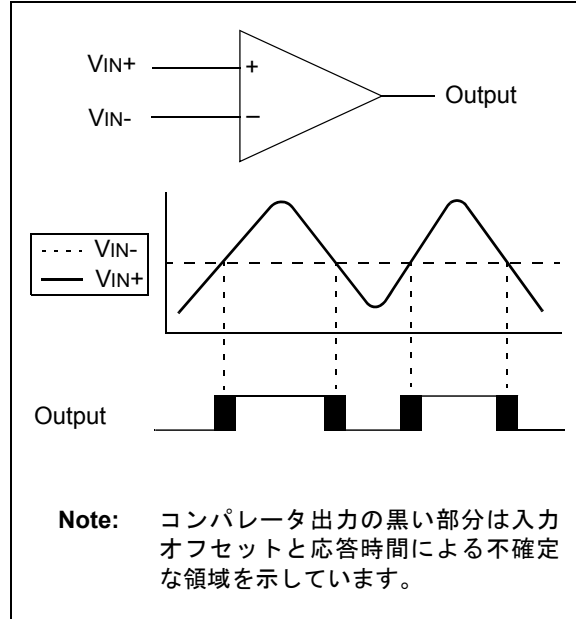
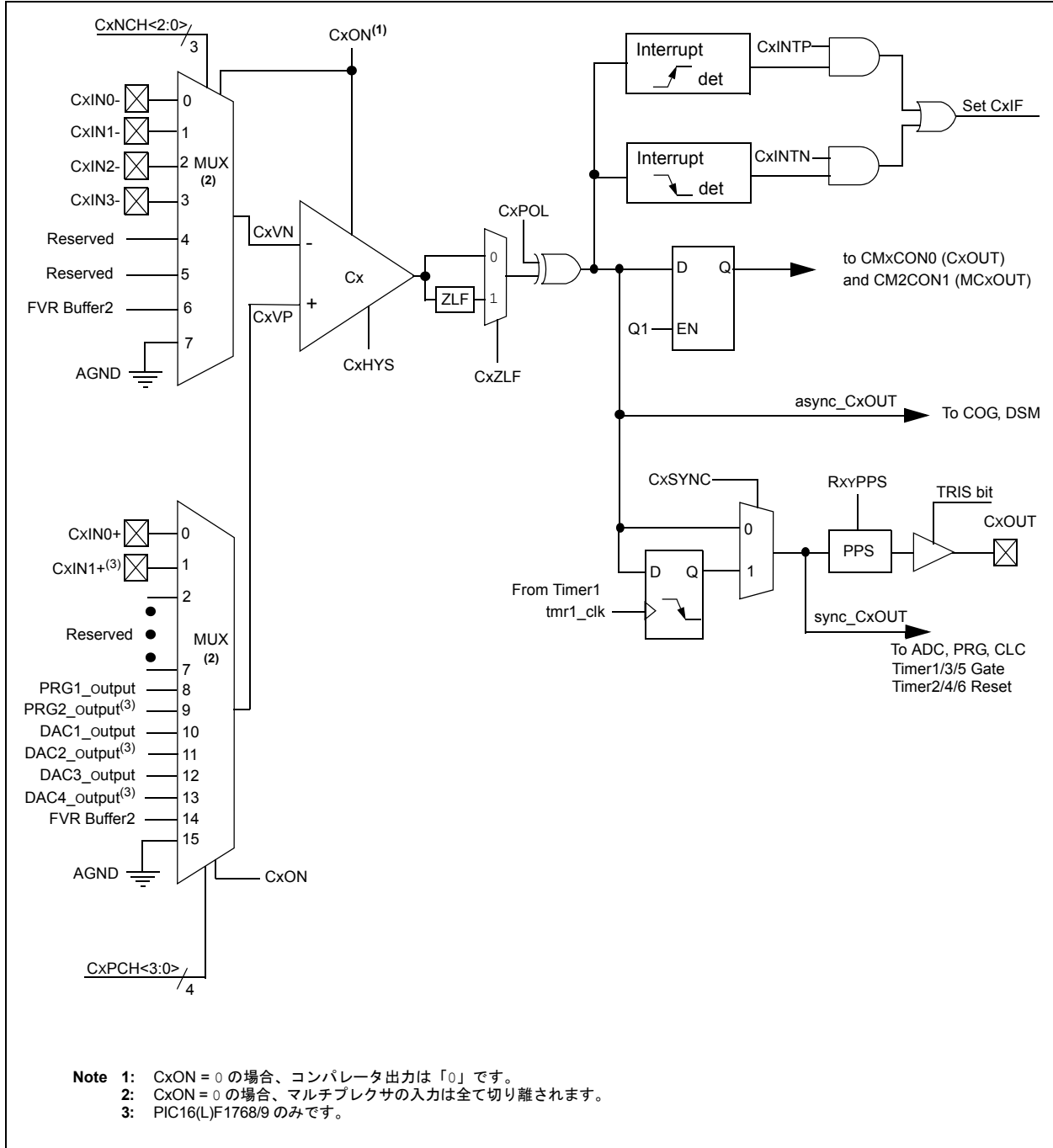


図 19-2: コンパレータ モジュールの概略ブロック図



PIC16(L)F1764/5/8/9

19.2 コンパレータ制御

各コンパレータには、2つの制御レジスタ (CMxCON0 と CMxCON1) があります。

CMxCON0 レジスタ (レジスタ 19-1 参照) には、以下に示す制御ビットとステータスビットがあります。

- イネーブル
- 出力
- 出力極性
- ゼロ遅延フィルタ
- 速度 / 消費電力の選択
- ヒステリシス イネーブル
- 出力同期

CMxCON1 レジスタ (レジスタ 19-2 参照) には、以下に示す制御ビットがあります。

- 割り込みイネーブル
- 割り込みエッジ極性
- 正入力チャンネル選択
- 負入力チャンネル選択

19.2.1 コンパレータの有効化

CMxCON0 レジスタの ON ビットをセットすると、コンパレータの動作が有効になります。ON ビットをクリアするとコンパレータが無効になり、消費電流は最小になります。

19.2.2 コンパレータ出力選択

コンパレータの出力は、CMxCON0 レジスタの OUT ビット、または CMOUT レジスタの MCxOUT ビットを読み出す事で監視できます。出力を外部に接続して使うには、以下の条件を満たす必要があります。

- 目的のピンが PPS でコンパレータ出力に設定されている
- 対応する TRIS ビットがクリアされている
- CMxCON0 レジスタの ON ビットがセットされている

Note 1: コンパレータの内部出力は各命令サイクルでラッチされます。特に指定のない限り、外部出力はラッチされません。

19.2.3 コンパレータ出力極性

コンパレータの出力を反転する事は、コンパレータの入力を入れ換える事と機能的に等価です。コンパレータ出力の極性を反転するには、CMxCON0 レジスタの POL ビットをセットします。POL ビットをクリアすると、非反転の出力になります。

表 19-2 に、入力条件と極性設定の組み合わせに対する出力状態を示します。

表 19-2: コンパレータの入力条件に対する出力状態

入力条件	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	1
$CxVN < CxVP$	1	0

19.3 コンパレータ ヒステリシス

コンパレータの入力ピンに一定の分離電圧を加える事で、ヒステリシス機能を持たせる事ができます。ヒステリシスは、CMxCON0 レジスタの HYS ビットをセットすると有効になります。

詳細は表 36-19: [コンパレータの仕様](#)を参照してください。

19.4 Timer1 のゲート動作

コンパレータの出力を Timer1 のゲート制御用の入力として使う事ができます。詳細は[セクション 22.6 「Timer1 ゲート」](#)を参照してください。この機能は、アナログイベントの継続期間または間隔を計測するのに使います。

コンパレータの出力は Timer1 に同期させる事を推奨します。これにより、コンパレータの値が変化している間にTimer1がインクリメントする事を防止できます。

19.4.1 コンパレータ出力の同期

CMxCON0 レジスタの SYNC ビットをセットすると、コンパレータの出力がTimer1 と同期します。

同期が有効になると、Timer1 クロック源の立ち下がりがエッジでコンパレータの出力がラッチされます。Timer1 でプリスケラを使っている場合、プリスケラ適用後のクロックに対してコンパレータ出力がラッチします。レーシングを回避するため、コンパレータ出力はTimer1クロック源の立ち上がりエッジでラッチされ、Timer1 はクロック源の立ち上がりエッジでインクリメントします。詳細は、コンパレータのブロック図 ([図 19-2](#)) と Timer1 のブロック図 ([図 22-1](#)) を参照してください。

19.5 コンパレータの割り込み

各コンパレータ出力値の変化に応じて割り込みを生成できます。立ち上がりおよび立ち下がりがエッジ検出機能を備えています。

どちらかのエッジが検出され、これに対応するイネーブルビット (CMxCON1 レジスタの INTP および INTN ビットのどちらかまたは両方) がセットされていると、対応する割り込みフラグビット (PIR2 レジスタの CxIF ビット) がセットされます。

割り込みを有効にするには、以下のビットをセットする必要があります。

- CMxCON0 レジスタの ON ビットと POL ビット
- PIE2 レジスタの CxIE ビット
- CMxCON1 レジスタの INTP ビット (立ち上がりエッジ検出の場合)
- CMxCON1 レジスタの INTN ビット (立ち下がりエッジ検出の場合)
- INTCON レジスタの PEIE および GIE ビット

対応する割り込みフラグビット (PIR2 レジスタの CxIF ビット) は、ソフトウェアでクリアする必要があります。このフラグをクリア中に次のエッジが検出された場合、シーケンス完了時にフラグはセットされたままです。

Note: コンパレータが無効の場合でも、CMxCON0 レジスタの POL ビットを使って出力極性を変更するか、CMxCON0 レジスタの ON ビットを使ってコンパレータのON/OFFを切り換える事で割り込みを生成できます。

19.6 コンパレータの正入力選択

コンパレータの非反転入力に接続する信号は、CMxCON1 レジスタの PCH<2:0> ビットで以下のいずれかを選択します。

- CxIN+ アナログピン
- プログラマブル ランプ ジェネレータ出力
- DAC 出力
- FVR(固定参照電圧)
- Vss(グランド)

固定参照電圧モジュールの詳細は、[セクション 14.0 「固定参照電圧 \(FVR\)」](#)を参照してください。

DAC 入力信号の詳細は、[セクション 17.0 「5 ビット D/Aコンバータ\(DAC\)モジュール」](#)を参照してください。

コンパレータが無効 (CxON = 0) の場合、全コンパレータ入力は常時無効です。

19.7 コンパレータの負入力選択

CMxCON0 レジスタの NCH<2:0> ビットを設定すると、以下に示すアナログ入力ピン、内部参照電圧、アナロググランドのいずれかがコンパレータの反転入力に接続されます。

- CxIN- ピン
- FVR(固定参照電圧)
- アナロググランド

反転入力に接続される信号には、オペアンプ出力とピンを共用するものがあります。両方の機能を同時に有効にすると、オペアンプ出力がコンパレータ反転入力に接続されます。

Note: CxINy+ ピンと CxINy- ピンをアナログ入力として使うには、ANSEL レジスタの対応するビットをセットすると共に、対応するTRISビットもセットして出力ドライバを無効にする必要があります。

PIC16(L)F1764/5/8/9

19.8 コンパレータの応答時間

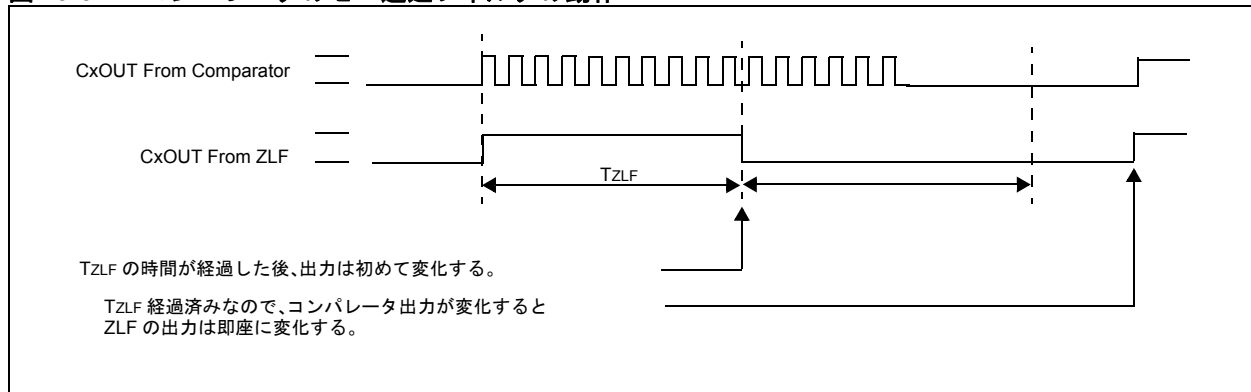
コンパレータの入力源または参照電圧を切り換えてから一定期間は、コンパレータ出力が不定となります。この時間を応答時間と呼びます。コンパレータの応答時間は、参照電圧のセトリングタイムとは別のものです。従って、コンパレータの入力が変化した時の総応答時間を求めるには、これら 2 つの時間を両方共考慮する必要があります。詳細は、表 36-19: コンパレータの仕様のコンパレータおよび参照電圧の仕様を参照してください。

とソフトウェアに悪影響を与えます。このため、コンパレータ出力に発振を抑制するデジタルフィルタを追加しました。このフィルタは、コンパレータ出力が変化してから 20 ns(公称値)の間、出力が再び反転するのを防ぎます。こうすることで、他のデバイスに悪影響を与えずにコンパレータ出力を安定させる事ができます。図 19-3 を参照してください。

19.9 ゼロ遅延フィルタ

高速モードでの動作中は、回路条件が適切であってもしきい値付近でコンパレータ出力が発振する可能性があります。この発振は、この信号を使うハードウェア

図 19-3: コンパレータのゼロ遅延フィルタの動作



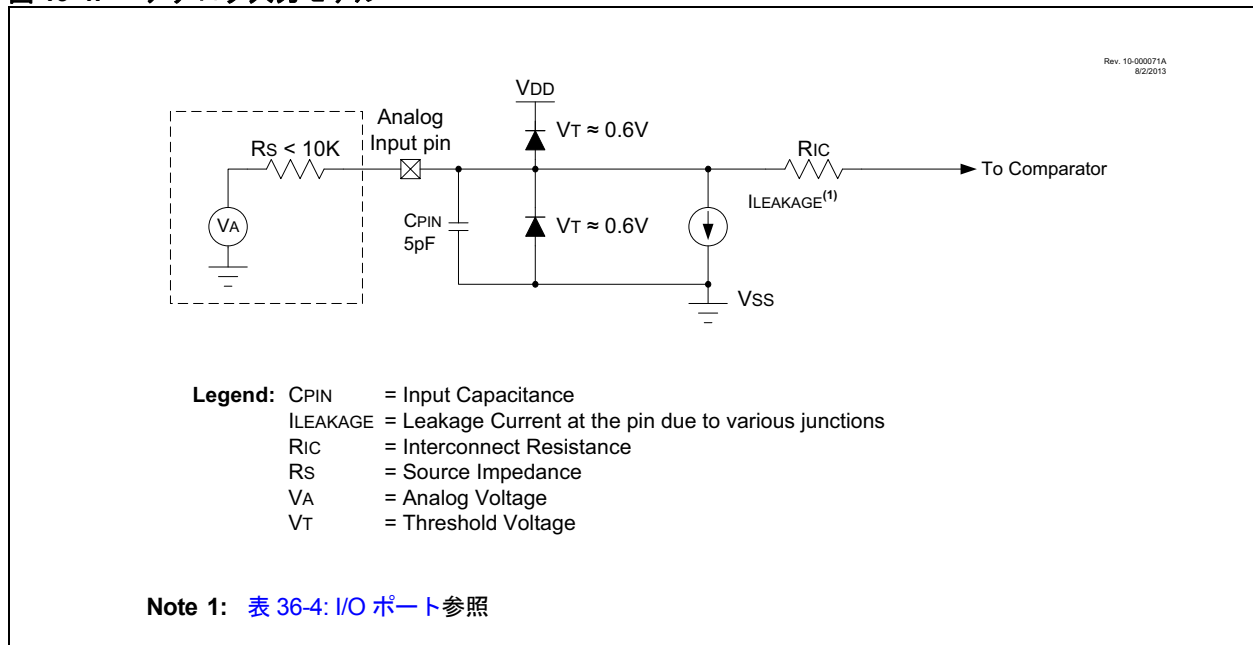
19.10 アナログ入力接続に関する注意事項

図 19-4 に、簡略化したアナログ入力回路を示します。アナログ入力ピンはデジタル入力と同じ接続を共有しているため、VDD と VSS には逆バイアスの ESD 保護ダイオードが接続されています。従って、アナログ入力は VSS と VDD の間の値である必要があります。入力電圧がこのレンジから $\pm 0.6\text{V}$ を超えて逸脱すると、片方のダイオードに順バイアスがかかり、ラッチアップが発生する事があります。

アナログ信号源の最大ソース インピーダンスは $10\text{ k}\Omega$ を推奨します。また、アナログ入力ピンにコンデンサまたはツェナー ダイオード等の外付け部品を接続する場合、精度の低下を最小限に抑えるためになるべくリーク電流の小さいものを使います。

- Note 1:** アナログ入力として設定された全てのピンは、PORT レジスタの読み出し時に「0」として読み出されます。デジタル入力として設定したピンは、入力仕様に基づいてアナログ入力に変換されます。
- 2:** デジタル入力として定義されたピンにアナログ電圧を加えると、入力バッファに仕様を超える電流が流れる場合があります。

図 19-4: アナログ入力モデル



PIC16(L)F1764/5/8/9

19.11 レジスタ定義：コンパレータ制御

表 19-3 に DSM 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1.2.2「完全ビット名」](#) を参照してください。

表 19-3:

周辺モジュール	ビット名の接頭辞
コンパレータ 1	C1
コンパレータ 2 ⁽¹⁾	C2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 19-1: CMxCON0: コンパレータ Cx 制御レジスタ 0

R/W-0/0	R-0/0	U-0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-0/0
ON	OUT	—	POL	ZLF	Reserved	HYS	SYNC
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **ON:** コンパレータ イネーブルビット
1 = コンパレータを有効にする
0 = コンパレータを無効にする (消費電力ゼロ)
- bit 6 **OUT:** コンパレータ出力ビット
POL = 1(極性反転)の場合:
1 = CxVP < CxVN
0 = CxVP > CxVN
POL = 0(極性非反転)の場合:
1 = CxVP > CxVN
0 = CxVP < CxVN
- bit 5 **未実装:** 「0」として読み出し
- bit 4 **POL:** コンパレータ出力極性選択ビット
1 = コンパレータ出力を反転する
0 = コンパレータ出力を反転しない
- bit 3 **ZLF:** コンパレータ ゼロ遅延フィルタ イネーブルビット
1 = コンパレータ出力にフィルタを適用する
0 = コンパレータ出力にフィルタを適用しない
- bit 2 **予約済み:** 「1」として読み出しこのビットはセットしたままにします。
- bit 1 **HYS:** コンパレータ ヒステリシス イネーブルビット
1 = コンパレータのヒステリシス機能を有効にする
0 = コンパレータのヒステリシス機能を無効にする
- bit 0 **SYNC:** コンパレータ出力同期モードビット
1 = Timer1 と I/O ピンに対するコンパレータ出力を Timer1 クロック源の変化に同期させる。出力は Timer1 クロック源の立ち下がリエッジで更新される
0 = Timer1 と I/O ピンに対するコンパレータ出力を非同期とする

レジスタ 19-2: CMxCON1: コンパレータ Cx 制御レジスタ 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-2 **未実装:** 「0」として読み出し

bit 1 **INTP:** 立ち上がりエッジでのコンパレータ割り込みイネーブルビット
1 = CxOUT ビットの立ち上がりエッジで CxIF 割り込みフラグをセットする
0 = CxOUT ビットの立ち上がりエッジで割り込みフラグをセットしない

bit 0 **INTN:** 立ち下がりエッジでのコンパレータ割り込みイネーブルビット
1 = CxOUT ビットの立ち下がりエッジで CxIF 割り込みフラグをセットする
0 = CxOUT ビットの立ち下がりエッジで割り込みフラグをセットしない

レジスタ 19-3: CMxNSEL: コンパレータ Cx 負チャンネル選択レジスタ

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	NCH<2:0>		
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-3 **未実装:** 「0」として読み出し

bit 2-0 **NCH<2:0>:** コンパレータ負入力チャンネル選択ビット
111 = CxVN に AGND を接続する
110 = CxVN に FVR バッファ 2 を接続する
101 = CxVN に PRG2_output⁽¹⁾ を接続する
100 = CxVN に PRG1_output を接続する
011 = CxVN に CxIN3- ピンを接続する
010 = CxVN に CxIN2- ピンを接続する
001 = CxVN に CxIN1- ピンを接続する
000 = CxVN に CxIN0- ピンを接続する

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 19-4: CMxPSEL: コンパレータ Cx 正チャンネル選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	PCH<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **PCH<3:0>:** コンパレータ正入力チャンネル選択ビット
1111 = CxVP に AGND を接続する
1110 = CxVP に FVR バッファ 2 を接続する
1101 = CxVP に DAC4_output を接続する
1100 = CxVP に DAC3_output を接続する
1011 = CxVP に DAC2_output を接続する
1010 = CxVP に DAC1_output を接続する
1001 = CxVP に PRG2_output を接続する
1000 = CxVP に PRG1_output を接続する
0111 = CxVP に何も接続しない。入力はフローティング
0110 = CxVP に何も接続しない。入力はフローティング
0101 = CxVP に何も接続しない。入力はフローティング
0100 = CxVP に何も接続しない。入力はフローティング
0011 = CxVP に何も接続しない。入力はフローティング
0010 = CxVP に何も接続しない。入力はフローティング
0001 = CxVP に CxIN0+ ピンを接続する
0000 = CxVP に CxIN0+ ピンを接続する

Note: 以下のミラーレジスタについては、完全ビット名と短縮ビット名はありません。

レジスタ 19-5: CMOUT: コンパレータ出力レジスタ

U-0	U-0	U-0	U-0	R-0/0	R-0/0	R-0/0	R-0/0
—	—	—	—	MC4OUT ⁽¹⁾	MC3OUT ⁽¹⁾	MC2OUT	MC1OUT
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-2 **未実装:** 「0」として読み出し
bit 3 **MC4OUT:** C4OUT ビットのミラーコピー⁽¹⁾
bit 2 **MC3OUT:** C3OUT ビットのミラーコピー⁽¹⁾
bit 1 **MC2OUT:** C2OUT ビットのミラーコピー
bit 0 **MC1OUT:** C1OUT ビットのミラーコピー

Note 1: PIC16(L)F1768/9 のみです。

表 19-4: コンパレータ モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
CM1CON0	ON	OUT	—	POL	ZLF	Reserved	HYS	SYNC	208
CM2CON0	ON	OUT	—	POL	ZLF	Reserved	HYS	SYNC	208
CM3CON0 ⁽¹⁾	ON	OUT	—	POL	ZLF	Reserved	HYS	SYNC	208
CM4CON0 ⁽¹⁾	ON	OUT	—	POL	ZLF	Reserved	HYS	SYNC	208
CM1CON1	—	—	—	—	—	—	INTP	INTN	209
CM2CON1	—	—	—	—	—	—	INTP	INTN	209
CM3CON1 ⁽¹⁾	—	—	—	—	—	—	INTP	INTN	209
CM4CON1 ⁽¹⁾	—	—	—	—	—	—	INTP	INTN	209
CM1NSEL	—	—	—	—	—	NCH<2:0>			209
CM2NSEL	—	—	—	—	—	NCH<2:0>			209
CM3NSEL ⁽¹⁾	—	—	—	—	—	NCH<2:0>			209
CM4NSEL ⁽¹⁾	—	—	—	—	—	NCH<2:0>			209
CM1PSEL	—	—	—	—	PCH<3:0>				210
CM2PSEL	—	—	—	—	PCH<3:0>				210
CM3PSEL ⁽¹⁾	—	—	—	—	PCH<3:0>				210
CM4PSEL ⁽¹⁾	—	—	—	—	PCH<3:0>				210
CMOUT	—	—	—	—	MC4OUT ⁽¹⁾	MC3OUT ⁽¹⁾	MC2OUT	MC1OUT	210
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173
DAC1CON0	EN	FM	OE1	—	PSS<1:0>		—	NSS	199
DAC2CON0 ⁽¹⁾	EN	FM	OE1	—	PSS<1:0>		—	NSS	199
DAC3CON0	EN	—	OE1	—	PSS<1:0>		—	NSS	193
DAC4CON0 ⁽¹⁾	EN	—	OE1	—	PSS<1:0>		—	NSS	193
DAC3REF	—	—	—	REF<4:0>					193
DAC4REF ⁽¹⁾	—	—	—	REF<4:0>					193
DAC1REFH	REF<9:x> (x は FM ビットによって異なる)								200
DAC2REFH ⁽¹⁾	REF<9:x> (x は FM ビットによって異なる)								200
DAC1REFL	REF<x-1:0> (x は FM ビットによって異なる)								200
DAC2REFL ⁽¹⁾	REF<x-1:0> (x は FM ビットによって異なる)								200
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例: — = 未実装、「0」として読み出し。網掛けの部分はコンパレータ モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

20.0 ゼロクロス検出(ZCD)モジュール

ZCD モジュールは、A/C 信号がグランド電位と交差するタイミングを検出します。実際のゼロクロスしきい値はゼロクロス参照電圧 ZCPINV で、この電圧はグランド + 0.75 V (typ.) です。

検出対象信号へは、直列電流制限抵抗を介して接続します。このモジュールは、ZCD ピンに電流を印加して (ソースまたはシンク) ピンの電圧を一定に保ち、ピンの電圧が ESD 保護ダイオードに順バイアスをかけないようにします。印加された電圧が参照電圧より高い場合、モジュールはシンク電流を流します。印加された電圧が参照電圧より低い場合、モジュールはソース電流を流します。このソースおよびシンク電流により、印加電圧レンジ全体でピンの電圧を一定に保ちます。図 20-2 に、ZCD モジュールの概略ブロック図を示します。

ZCD モジュールは、以下の目的での AC 波形監視時に役立ちますが、これらの目的に限定される訳ではありません。

- A/C 周期の計測
- 正確な長時間の計測
- 調光位相遅延ドライブ
- 低 EMI サイクルスイッチング

20.1 外付け抵抗の選択

ZCD モジュールには、外部電圧源と直列に接続した電流制限抵抗が必要です。この抵抗のインピーダンスと定格は、外部電源のピーク電圧を考慮して求めます。抵抗値は、抵抗を流れる電流が公称 300 μ A の時に全てのピーク電圧を降下できるように選択します (式 20-1 と図 20-1 参照)。電流のソースおよびシンクに干渉しないよう、ZCD I/O ピンの内部弱プルアップを必ず無効にします。

式 20-1: 外付け抵抗

$$R_{series} = \frac{V_{peak}}{3 \times 10^{-4}}$$

図 20-1: 外部電圧

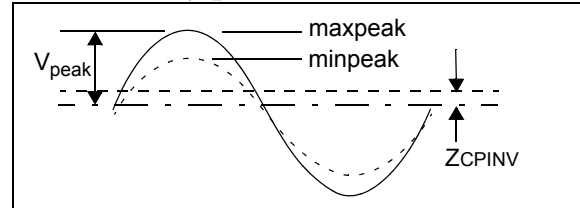
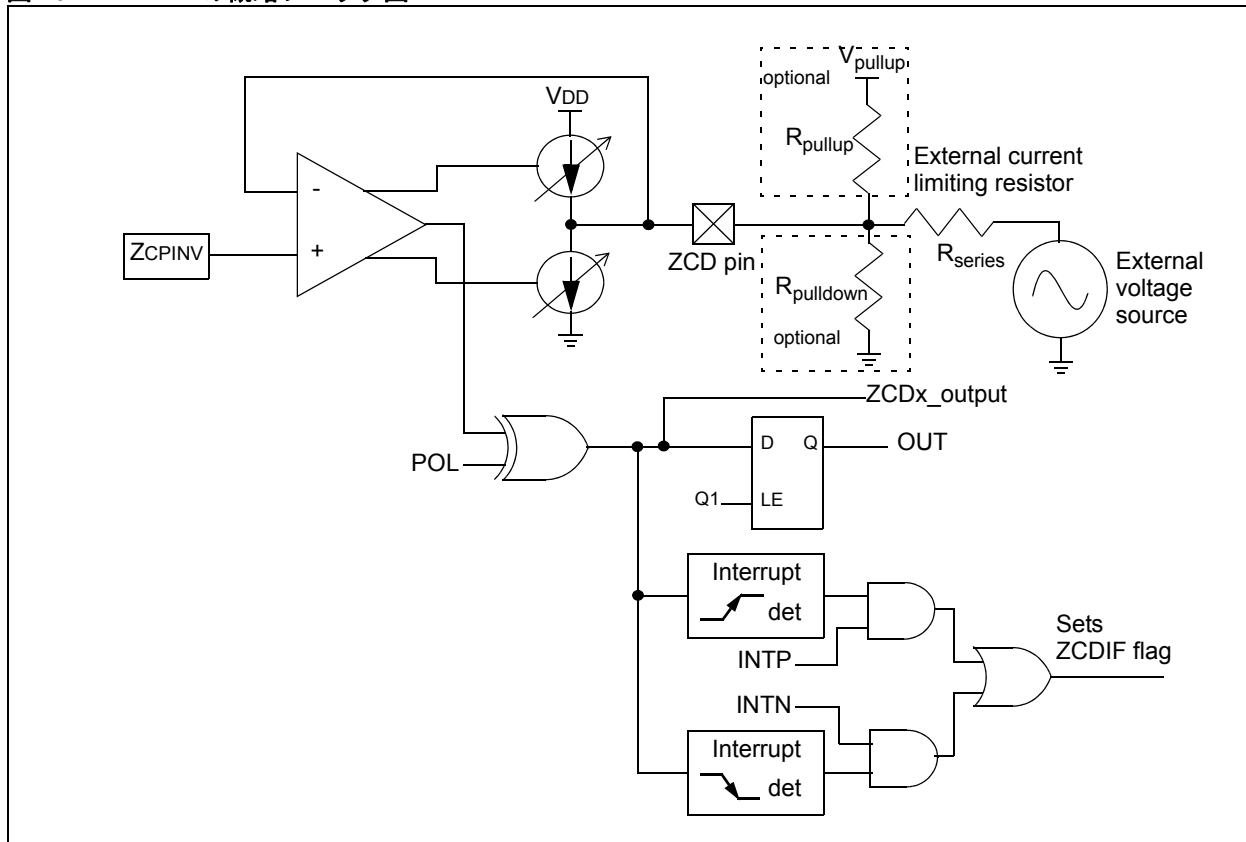


図 20-2: ZCD の概略ブロック図



20.2 ZCD ロジック出力

ZCD モジュールはステータスビットを備えています。このビットを読み出すと、電流のソースとシンクどちらがアクティブであるか調べられます。ZCDxCON レジスタの OUT ビットはシンク時にセット、ソース時にクリアされます。OUT ビットは極性ビットの影響を受けます。

20.3 ZCD ロジック極性

ZCDxCON レジスタの POL ビットは、電流のソースとシンクに対応する OUT ビットを反転します。POL ビットがセットされている時に OUT が High であればソース、Low であればシンク中である事を示します。

POL ビットは ZCD 割り込みに影響を与えます。[セクション 20.4「ZCD 割り込み」](#)を参照してください。

20.4 ZCD 割り込み

対応する割り込みイネーブルをセットすると、ZCD ロジック出力の変化に応じて割り込みが生成されます。この目的のために、ZCD は立ち上がりエッジ検出機能と立ち下がりエッジ検出機能を備えています。

どちらかのエッジ検出機能がトリガされ、対応するイネーブルビットがセットされている場合、PIR3 レジスタの ZCDIF ビットがセットされます。INTP ビットは立ち上がりエッジ割り込みを、INTN ビットは立ち下がりエッジ割り込みを有効にします。どちらのビットも ZCDxCON レジスタにあります。

この割り込みを完全に有効にするには、以下のビットをセットする必要があります。

- PIE3 レジスタの ZCDIE ビット
- ZCDxCON レジスタの INTP ビット (立ち上がりエッジ検出の場合)
- ZCDxCON レジスタの INTN ビット (立ち下がりエッジ検出の場合)
- INTCON レジスタの PEIE および GIE ビット

POL ビットを変化させると、EN ビットの状態に関係なく割り込みが発生します。

PIR3 レジスタの ZCDIF ビットは、割り込みサービスの一部としてソフトウェアでクリアする必要があります。このフラグをクリア中に次のエッジが検出された場合、シーケンス完了時にフラグはセットされたままです。

20.5 ZCPINV オフセットの補正

ZCD は、実際には ZCD オペアンプの非反転入力に接続された参照電圧レベルでスイッチングします。外部電圧源が矩形波以外の波形の場合、ゼロからの電圧オフセットによってゼロクロス イベントの発生が早過ぎるか遅過ぎるかのどちらかになります。波形が V_{ss} に対して変動している場合、波形の立ち下がりでゼロクロス検出が早くなり、波形の立ち上がりでは遅くなります。波形が V_{DD} に対して変動している場合、波形の立ち上がりでゼロクロス検出が遅くなり、波形の立ち下がりでは早くなります。正弦波の波形の場合の実際のオフセット時間は、[式 20-2](#) でそれぞれ求める事ができます。

式 20-2: ZCD イベントのオフセット

外部電圧源が V_{ss} に対して変動している場合：

$$T_{offset} = \frac{\text{asin}\left(\frac{Z_{cpinv}}{V_{peak}}\right)}{2\pi \cdot Freq}$$

外部電圧源が V_{DD} に対して変動している場合：

$$T_{offset} = \frac{\text{asin}\left(\frac{V_{DD}-Z_{cpinv}}{V_{peak}}\right)}{2\pi \cdot Freq}$$

このオフセット時間は、プルアップまたはプルダウンバイアス抵抗を ZCD ピンに接続する事で補償できます。外部電圧源が V_{ss} に対して変動している場合はプルアップ抵抗を使います。電圧が V_{DD} に対して変動している場合はプルダウン抵抗を使います。この抵抗によって ZCD ピンにバイアスが追加され、ターゲットの外部電圧源がゼロにならないとピン電圧をスイッチング電圧 Z_{CPINV} まで引き上げる事ができなくなります。プルアップまたはプルダウンの値は[式 20-3](#) で求める事ができます。

式 20-3: ZCD のプルアップ/プルダウン

外部信号が V_{ss} に対して変動している場合：

$$R_{pullup} = \frac{R_{series}(V_{pullup}-Z_{cpinv})}{Z_{cpinv}}$$

外部信号が V_{DD} に対して変動している場合：

$$R_{pulldown} = \frac{R_{series}(Z_{cpinv})}{(V_{DD}-Z_{cpinv})}$$

PIC16(L)F1764/5/8/9

プルアップおよびプルダウン抵抗の値は ZCPINV がわずかに変動するだけで大きく変化します。ZCPINV の計測は、特に波形が VDD に対して変動している場合は困難です。しかし、式 20-2 と式 20-3 を組み合わせると、ZCDOUT の High 期間と Low 期間の長さの差から抵抗の値を求める事ができます。なお、この期間の差 ΔT は、 $4 \cdot T_{\text{offset}}$ です。式 20-4 に、ZCDOUT の High 期間と Low 期間からプルアップおよびプルダウン抵抗の値を求める式を示します。ZCDOUT 信号は、いずれかの CLC に接続するとピンに出力して直接監視できます。

式 20-4:

$$R = R_{\text{series}} \left(\frac{V_{\text{bias}}}{V_{\text{peak}} \left(\sin \left(\pi \text{Freq} \frac{\Delta T}{2} \right) \right)} - 1 \right)$$

R: プルアップまたはプルダウン抵抗の値

V_{bias} : V_{pullup} (R がプルアップの場合)
または VDD (R がプルダウンの場合)

ΔT : ZCDOUT の High 期間と Low 期間の差

20.6 V_{peak} の変動への対処

外部電圧源のピーク振幅に変動が予測される場合、ZCD 電流ソースとシンクが予測される最大電圧の時に回路設計の最大レンジである $\pm 600 \mu\text{A}$ 以内に収まるようにすると共に、ピーク電圧が最小の時にも正確に検出できる大きさとなるように直列抵抗を選択する必要があります。経験則として、最大ピーク電圧は最小ピーク電圧の 6 倍以内とします。最大電流が $\pm 600 \mu\text{A}$ を超えず、なおかつ最小電流が $\pm 100 \mu\text{A}$ 以上となるようにするには、式 20-5 で直列抵抗の値を求めます。プルアップの値はピーク電圧から独立しているため、この直列抵抗に対する補償プルアップの値は、式 20-3 で求める事ができます。

式 20-5: 電圧レンジに対する直列抵抗

$$R_{\text{series}} = \frac{V_{\text{maxpeak}} + V_{\text{minpeak}}}{7 \times 10^{-4}}$$

20.7 スリープ中の動作

ZCD の電流源と割り込みはスリープの影響を受けません。

20.8 リセットの影響

ZCD 回路は、パワーオンリセット (POR) 時の既定値 (有効 / 無効) を設定できます。ZCD コンフィグレーションビットがクリアの場合、ZCD 回路は POR 時に有効です。ZCD コンフィグレーションビットがセットされている場合、ZCDxCON レジスタの ZCDxEN ビットをセットするまで ZCD モジュールは無効です。

20.9 レジスタ定義 :ZCD 制御

表20-1にゼロクロス検出周辺モジュールの完全ビット名の接頭辞を示します。詳細は 1.1.2.2 「完全ビット名」を参照してください。

表 20-1:

周辺モジュール	ビット名の接頭辞
ZCD1	ZCD1

レジスタ 20-1: ZCDxCON: ゼロクロス検出制御レジスタ

R/W-0/0	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
EN	—	OUT	POL	—	—	INTP	INTN
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

- bit 7 **EN:** ゼロクロス検出イネーブルビット (1)
 1 = ゼロクロス検出を有効にする。ZCD ピンからソースまたはシンク電流を出力する
 0 = ゼロクロス検出を無効にする。ZCD ピンは PPS および TRIS 制御に従って動作する
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **OUT:** ゼロクロス検出論理レベルビット
POL ビット = 0:
 1 = ZCD ピンはシンク電流を流している
 0 = ZCD ピンはソース電流を流している
POL ビット = 1:
 1 = ZCD ピンはソース電流を流している
 0 = ZCD ピンはシンク電流を流している
- bit 4 **POL:** ゼロクロス検出論理出力極性ビット
 1 = ZCD 論理出力を反転する
 0 = ZCD 論理出力を反転しない
- bit 3-2 **未実装:** 「0」として読み出し
- bit 1 **INTP:** ゼロクロス立ち上がりエッジ割り込みイネーブルビット
 1 = OUT が Low から High に遷移した時に ZCDIF ビットをセットする
 0 = OUT が Low から High に遷移しても ZCDIF ビットをセットしない
- bit 0 **INTN:** ゼロクロス立ち下がりエッジ割り込みイネーブルビット
 1 = OUT が High から Low に遷移した時に ZCDIF ビットをセットする
 0 = OUT が High から Low に遷移しても ZCDIF ビットをセットしない

Note 1: $\overline{\text{ZCD}}$ コンフィグレーション ビットがクリアの場合、EN ビットは無視されます。

PIC16(L)F1764/5/8/9

表 20-2: ZCD モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ内容記載ページ
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
ZCD1CON	EN	—	OUT	POL	—	—	INTP	INTN	215

凡例: — = 未実装、「0」として読み出し。網掛けの部分は ZCD モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

表 20-3: ZCD モジュール関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ内容記載ページ
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	PLLEN	67
	7:0	$\overline{\text{ZCD}}$	—	—	—	—	PPS1WAY	WRT<1:0>		

凡例: — = 未実装、「0」として読み出し。網掛けの部分は ZCD モジュールでは使いません。

21.0 Timer0 モジュール

Timer0 モジュールは 8 ビットのタイマ / カウンタで、以下の特長を備えています。

- 8 ビットのタイマ / カウンタレジスタ (TMR0)
- 8 ビットのプリスケアラ (ウォッチドッグ タイマから独立)
- プログラマブルな内部または外部クロック源
- プログラマブルな外部クロックエッジ選択
- オーバーフロー割り込み
- TMR0 は Timer1 のゲート制御に使用可能

図 21-1 に、Timer0 モジュールのブロック図を示します。

21.1 Timer0 の動作

Timer0 モジュールは、8 ビットのタイマまたはカウンタとして使えます。

21.1.1 8 ビットタイマ モード

プリスケアラを使わない場合、Timer0 モジュールは 1 命令サイクルごとにインクリメントします。8 ビットタイマ モードを選択するには、OPTION_REG レジスタの TMR0CS ビットをクリアします。

TMR0 に書き込みが実行されると、書き込み直後の 2 命令サイクル間はインクリメントが抑止されます。

Note: TMR0 への書き込み時に生じる 2 命令サイクル間の遅延は、TMR0 レジスタに書き込む値を調整する事で補償できます。

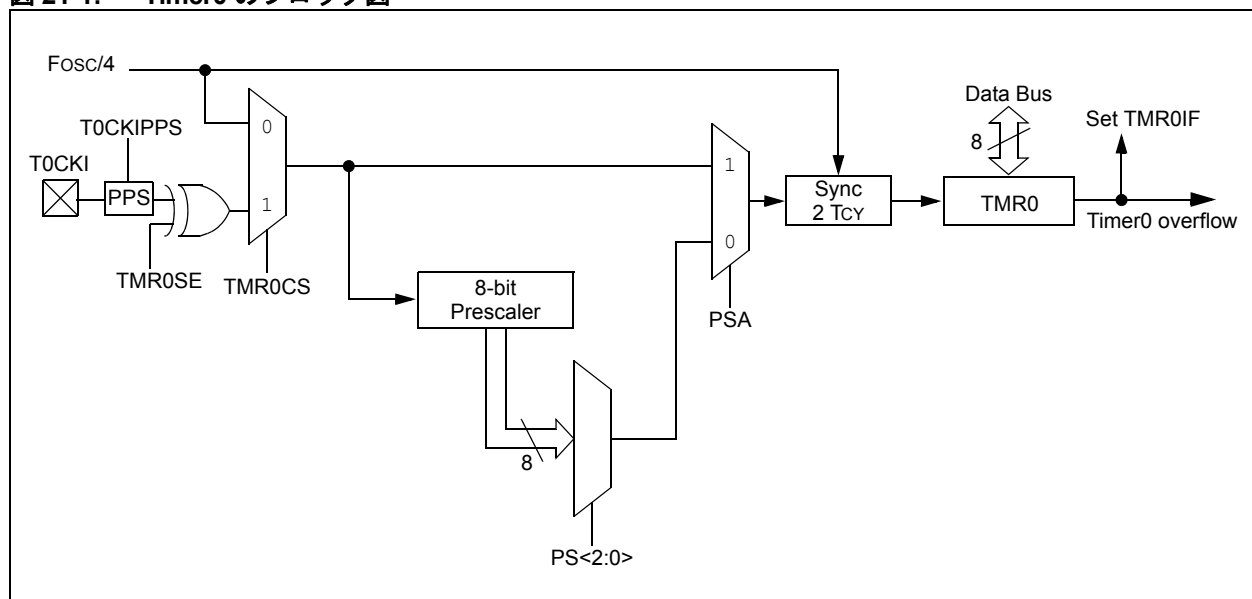
21.1.2 8 ビットカウンタ モード

8 ビットカウンタ モードでは、Timer0 モジュールは T0CKI ピンの立ち上がりエッジまたは立ち下がりエッジごとにインクリメントします。

T0CKI ピンを使った 8 ビットカウンタ モードを選択するには、OPTION_REG レジスタの TMR0CS ビットを「1」にセットします。

立ち上がりまたは立ち下がりエッジのどちらでインクリメントするかは、OPTION_REG レジスタの TMR0SE ビットで選択します。

図 21-1: Timer0 のブロック図



PIC16(L)F1764/5/8/9

21.1.3 ソフトウェアでプログラム可能なプリスケアラ

ソフトウェアでプログラム可能なプリスケアラは Timer0 専用です。このプリスケアラを有効にするには、OPTION_REG レジスタの PSA ビットをクリアします。

Note: ウォッチドッグタイマ(WDT)は専用のプリスケアラを使います。

Timer0 モジュールのプリスケアラは、1:2 から 1:256 までの 8 つの設定が可能です。プリスケアラ値は OPTION_REG レジスタの PS<2:0> ビットで選択します。Timer0 モジュールに 1:1 のプリスケアラ値を設定するには、OPTION_REG レジスタの PSA ビットをセットしてプリスケアラを無効にします。

プリスケアラ値の読み書きはできません。TMR0 レジスタに対する全ての書き込み命令は、プリスケアラをクリアします。

21.1.4 Timer0 の割り込み

TMR0 レジスタが FFh から 00h にオーバーフローすると、Timer0 は割り込みを生成します。Timer0 割り込みの有効 / 無効にかかわらず、TMR0 レジスタがオーバーフローするたびに、INTCON レジスタの TMR0IF 割り込みフラグビットがセットされます。TMR0IF ビットは、ソフトウェアでのみクリアできます。Timer0 割り込みを有効にするには、INTCON レジスタの TMR0IE ビットをセットします。

Note: スリープ中は Timer0 が停止しているため、このタイマの割り込みによってプロセッサをスリープから復帰させる事はできません。

21.1.5 8 ビットカウンタ モードの同期

8 ビットカウンタ モードの場合、TOCKI ピンに印加されるインクリメント エッジは命令クロックに同期している必要があります。同期を確立するには、命令クロックの Q2 と Q4 のサイクルでプリスケアラ出力をサンプリングします。外部クロック源の High/Low 期間は、表 36-12: Timer0 と Timer1 の外部クロック要件に記載されたタイミング要件を満たす必要があります。

21.1.6 スリープ中の動作

プロセッサがスリープ中、Timer0 は動作しません。プロセッサがスリープ中、TMR0 レジスタの内容は変更されません。

21.2 レジスタ定義 : OPTION レジスタ

レジスタ 21-1: OPTION_REG: OPTION レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **WPUEN**: 弱プルアップ イネーブルビット
1 = 全ての弱プルアップを無効にする (MCLR が有効の場合、MCLR を除く)
0 = 各 WPUx ラッチの値に応じて弱プルアップを有効にする
- bit 6 **INTEDG**: 割り込みエッジ選択ビット
1 = INT ピンの立ち上がりエッジで割り込み
0 = INT ピンの立ち下がりエッジで割り込み
- bit 5 **TMR0CS**: Timer0 クロック源選択ビット
1 = T0CKI ピンの遷移
0 = 内部命令サイクルクロック (Fosc/4)
- bit 4 **TMR0SE**: Timer0 信号源エッジ選択ビット
1 = T0CKI ピンの High から Low への遷移時にインクリメントする
0 = T0CKI ピンの Low から High への遷移時にインクリメントする
- bit 3 **PSA**: プリスケアラ割り当てビット
1 = プリスケアラを Timer0 モジュールに割り当てない
0 = プリスケアラを Timer0 モジュールに割り当てる
- bit 2-0 **PS<2:0>**: プリスケアラ比選択ビット

ビット値	Timer0 比
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 21-1: Timer0 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
TMR0	Timer0 モジュール レジスタ								217*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139

凡例: — = 未実装、「0」として読み出し。網掛け部分は Timer0 モジュールでは使用しません。

* 本文中でレジスタを説明しているページです。

Note 1: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

22.0 Timer1/3/5 モジュール (ゲート制御対応)

Timer1 モジュールは 16 ビットのタイマ/カウンタで、以下の特長を備えています。

- 16 ビットのタイマ/カウンタ レジスタペア (TMR1H:TMR1L)
- プログラブルな内部または外部クロック源
- 2 ビットプリスケアラ
- 専用の 32 kHz オシレータ回路
- 同期コンパレータ出力 (オプション)
- 複数の Timer1 ゲート (カウントイネーブル) ソース
- オーバーフロー割り込み
- オーバーフローによる復帰 (外部クロック、非同期モードのみ)
- キャプチャ/コンペア機能のためのタイムベース

- (CCP による) 自動変換トリガ
- 選択可能なゲートソース極性
- ゲートトグルモード
- ゲートシングルパルスモード
- ゲート値のステータス
- ゲートイベントによる割り込み

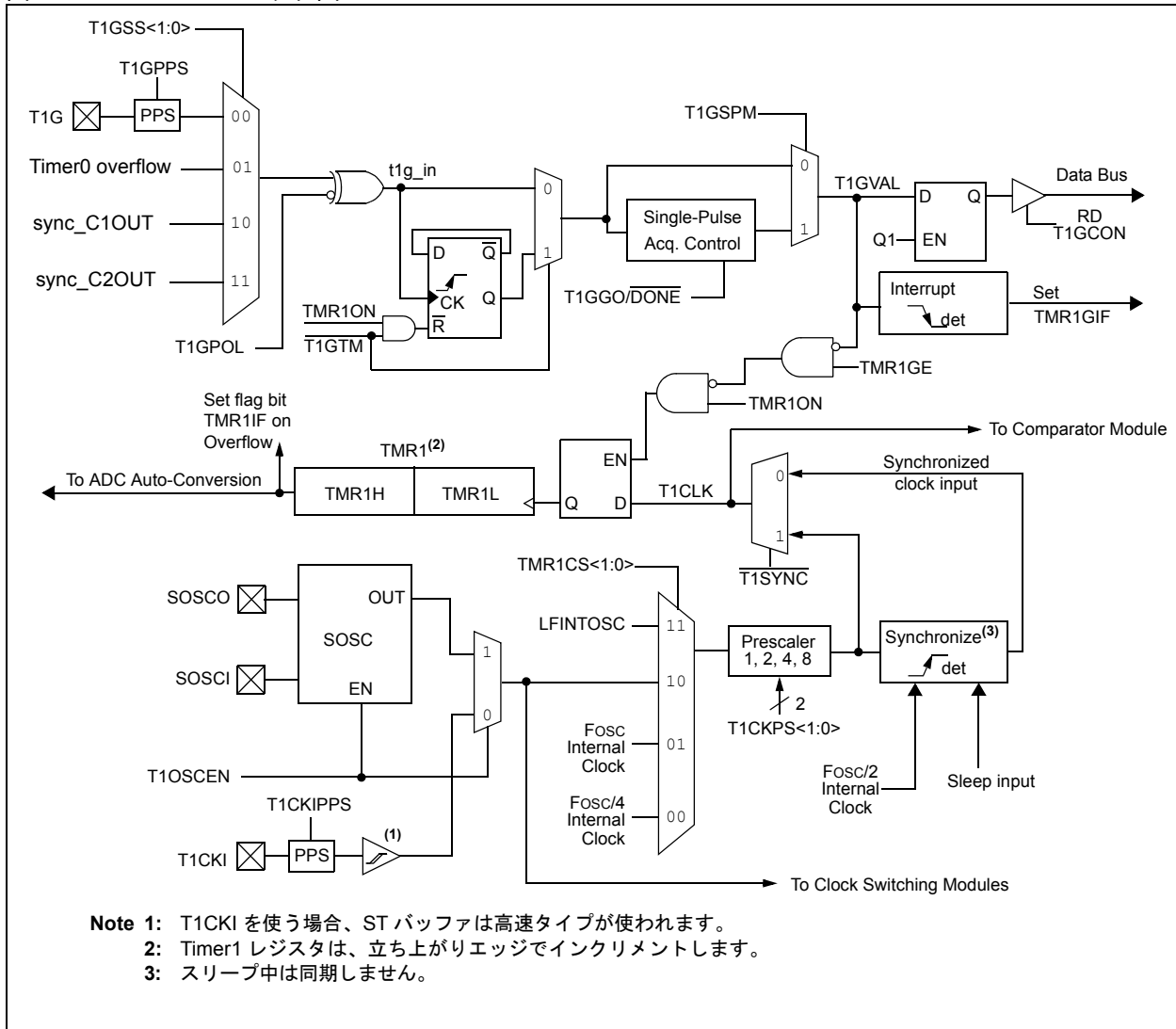
図 22-1 に、Timer1 モジュールのブロック図を示します。

デバイスは Timer1 タイプのモジュールとして以下の 3 つのインスタンスを持っています。

- Timer1
- Timer3
- Timer5

Timer1 と Timer1 ゲートに関する説明は全て Timer3 と Timer5 にも該当します。

図 22-1: Timer1 ブロック図



22.1 Timer1 の動作

Timer1 モジュールは 16 ビットのインクリメントカウンタで、TMR1H:TMR1L レジスタペアを介してアクセスします。TMR1H または TMR1L に書き込むと、カウンタ値を直接更新できます。

内部クロック源を使うと、このモジュールはタイマとして機能し、1 命令サイクルごとにインクリメントします。外部クロック源を使うと、このモジュールはタイマまたはカウンタとして機能し、外部クロック源の指定したエッジごとにインクリメントします。

Timer1 を有効にするには、T1CON レジスタの ON ビットと T1GCON レジスタの GE ビットをそれぞれセットします。表 22-1 に Timer1 のイネーブル選択を示します。

表 22-1: Timer1 イネーブル選択

TMR1ON	TMR1GE	Timer1 の動作
0	0	OFF
0	1	OFF
1	0	常に ON
1	1	カウント有効

22.2 クロック源の選択

T1CON レジスタの CS<1:0> および OSCEN ビットで Timer1 のクロック源を選択します。表 22-2 にクロック源の選択を示します。

22.2.1 内部クロック源

内部クロック源を選択した場合、TMR1H:TMR1L レジスタペアは、Timer1 プリスケアラで決まる Fosc の倍数単位でインクリメントします。

Fosc の内部クロック源を選択した場合、Timer1 レジスタの値は、1 命令クロックサイクルあたり 4 カウント分インクリメントします。このため、Timer1 の値を読み出す場合、分解能に 2 LSB の誤差が生じます。Timer1 の分解能を最大限に活かすには、Timer1 クロック入力をゲート制御する非同期の入力信号が必要です。

非同期の信号源として、以下のものが使えます。

- Timer1 ゲートへ接続される T1G ピン上の非同期イベント
- Timer1 ゲートへの C1/C2 コンパレータ入力

22.2.2 外部クロック源

外部クロック源を選択した場合、Timer1 モジュールはタイマまたはカウンタとして動作します。

カウンタとして使う場合、Timer1 は外部クロック入力 T1CKI の立ち上がりエッジでインクリメントします。この外部クロック入力は、マイクロコントローラのシステムクロックに同期させる事が可能ですが、非同期で動作させる事もできます。

クロック オシレータによるタイマとして使う場合、専用の内部オシレータ回路と外付けの 32.768 kHz 水晶振動子を使えます。

Note: カウンタモードの場合、下記の 1 つまたは複数の条件発生後はカウンタが最初の立ち上がりエッジでインクリメントする前に、立ち下がりエッジを 1 回検出する必要があります。

- POR 後 Timer1 を有効にした
- TMR1H または TMR1L に書き込んだ
- Timer1 を無効にした
- T1CKI が High の時に Timer1 を無効 (TMR1ON = 0) にして、その後 T1CKI が Low の時に Timer1 を有効 (TMR1ON=1) にした

表 22-2: クロック源の選択

TMR1CS<1:0>	T1OSCEN	クロック源
11	x	LFINTOSC
10	0	T1CKI ピンの外部クロック
01	x	システムクロック (Fosc)
00	x	命令クロック (Fosc/4)

PIC16(L)F1764/5/8/9

22.3 Timer1 プリスケーラ

Timer1 のプリスケーラは、クロック入力の 1/1、1/2、1/4、1/8 倍の 4 種類の設定が可能です。プリスケーラカウンタは、T1CON レジスタの CKPS ビットで制御します。プリスケーラカウンタは直接読み書きできませんが、TMR1H または TMR1L への書き込み時にクリアされます。

22.4 Timer1(セカンダリ)オシレータ

専用の 32.768 kHz 低消費電力オシレータ回路が、SOSCI(入力)と SOSCO(アンプ出力)の間に内蔵されています。この内部回路は、外付けの 32.768 kHz 水晶振動子と組み合わせて使います。

オシレータ回路は、T1CON レジスタの OSCEN ビットをセットすると有効になります。このオシレータは、スリープ中でも動作を継続します。

Note: このオシレータを使う場合、起動と安定化の時間が必要です。従って、Timer1 を使うには OSCEN をセットし、適切な遅延を確保する必要があります。OST 遅延と同様の遅延をソフトウェアで実装するには、TMR1IF ビットをクリアしてから TMR1H:TMR1L レジスタペアを FC00h にプリセットします。1024 クロックサイクルが経過すると TMR1IF フラグがセットされ、オシレータが動作中でかつ十分安定している事を示します。

22.5 非同期カウンタモード時の Timer1 動作

T1CON レジスタの SYNC 制御ビットをセットした場合、外部クロック入力は同期されません。タイマは内部相クロックに非同期でインクリメントします。外部クロック源が選択されている場合、タイマはスリープ中でも動作を継続するため、オーバーフロー割り込みを生成してプロセッサを復帰させる事ができます。しかし、このタイマの読み書きには、ソフトウェアに特別な注意を払う必要があります(セクション22.5.1「非同期カウンタモードにおける Timer1 の読み書き」参照)。

Note: 動作を同期モードから非同期モードに切り換える際は、インクリメントがスキップされる可能性があります。非同期モードから同期モードに切り換える際は、インクリメントが 1 回多く実行される可能性があります。

22.5.1 非同期カウンタモードにおける Timer1 の読み書き

タイマが外部の非同期クロックで動作している間は、TMR1H または TMR1L の正しい読み出し動作が(ハードウェアによって)保証されます。しかし、16 ビットタイマを 2 つの 8 ビット値として読み出すため、2 回の読み出しの間にタイマがオーバーフローする可能性に注意する事が必要です。

タイマへの書き込みの場合、実行前にタイマを停止する事を推奨します。レジスタがインクリメント中にタイマレジスタへ書き込むと、書き込みの競合が発生する場合があります。この場合、TMR1H:TMR1L レジスタペアが予期しない値となる事があります。

22.6 Timer1 ゲート

Timer1 は、フリーランニングカウンタ、または Timer1 ゲート回路による有効/無効切り換えが可能なカウンタとして設定できます。この機能は Timer1 ゲートイネーブルとも呼ばれます。

Timer1 ゲートは複数の選択可能なソースによって駆動できます。

22.6.1 Timer1 ゲートイネーブル

Timer1 ゲートイネーブルモードを有効にするには、T1GCON レジスタの GE ビットをセットします。Timer1 ゲートイネーブルモードの極性は、T1GCON レジスタの GPOL ビットで設定します。

Timer1ゲートイネーブルモードが有効の場合、Timer1 は Timer1 クロック源の立ち上がりエッジでインクリメントします。Timer1 ゲートイネーブルモードが無効の場合、Timer1 はインクリメントせずに現在のカウンタ値を保持します。詳細なタイミングは図 22-3 を参照してください。

表 22-3: Timer1 ゲートイネーブルの選択

T1CLK	T1GPOL	T1G	Timer1 の動作
↑	0	0	カウントする
↑	0	1	カウントを保持する
↑	1	0	カウントを保持する
↑	1	1	カウントする

22.6.2 Timer1 ゲートのソース選択

表 22-4 に Timer1 ゲートのソース選択を示します。ソース選択には、T1GCON レジスタの T1GSS ビットを使います。使える各ソースの極性も選択できます。極性の選択には、T1GCON レジスタの T1GPOL ビットを使います。

表 22-4: Timer1 ゲートのソース

T1GSS	Timer1 ゲートのソース
00	Timer1 ゲートピン
01	Timer0 のオーバーフロー (TMR0の FFh から 00h へのインクリメント)
10	コンパレータ 1 出力 (sync_C1OUT) (オプションで Timer1 に同期した出力)
11	コンパレータ 2 出力 (sync_C2OUT) (オプションで Timer1 に同期した出力)

22.6.2.1 T1G ピンのゲート動作

T1G ピンは、Timer1 のゲート制御用に使われるソースの 1 つです。このピンから Timer1 のゲート回路に外部ソースを供給できます。

22.6.2.2 Timer0 オーバーフロー ゲート動作

Timer0 が FFh から 00h へインクリメントすると、Low から High に遷移するパルスが自動的に生成され、内部で Timer1 ゲート回路に供給されます。

22.6.2.3 コンパレータ C1 のゲート動作

Timer1 のゲート制御用ソースとして、コンパレータ 1 動作の結果出力を選択できます。コンパレータ 1 の出力 (sync_C1OUT) は、Timer1 のクロックに同期させるか、非同期のまま動作させる事が可能です。詳細は [セクション 19.4.1「コンパレータ出力の同期」](#) を参照してください。

22.6.2.4 コンパレータ C2 のゲート動作

Timer1 のゲート制御用ソースとして、コンパレータ 2 動作の結果出力を選択できます。コンパレータ 2 の出力 (sync_C2OUT) は、Timer1 のクロックに同期させるか、非同期のまま動作させる事が可能です。詳細は [セクション 19.4.1「コンパレータ出力の同期」](#) を参照してください。

22.6.3 Timer1 ゲートトグル モード

Timer1 のゲートトグル モードが有効の場合、シングルレベルパルスの長さではなく、Timer1 ゲート信号のフルサイクルの長さを計測できます。

Timer1 ゲートのソースは、信号のインクリメントエッジごとにステートが変化するフリップフロップを介して接続されます。詳細なタイミングは [図 22-4](#) を参照してください。

Timer1 ゲートトグル モードを有効にするには、T1GCON レジスタの T1GTM ビットをセットします。T1GTM ビットをクリアすると、フリップフロップはクリアされて、その状態を保持します。これは、どのエッジを計測するかを制御するために必要です。

Note: ゲート極性の変更と同時にトグルモードを有効にすると、不定動作が生じる可能性があります。

22.6.4 Timer1 ゲートシングルパルス モード

Timer1 ゲート シングルパルス モードを有効にすると、シングルパルスのゲートイベントをキャプチャできます。Timer1 ゲート シングルパルス モードを有効にするには、まず T1GCON レジスタの T1GSPM ビットをセットします。次に、T1GCON レジスタの T1GGO/DONE ビットをセットします。Timer1 は、次のインクリメントエッジで完全に有効になります。パルスの次のトレーリングエッジで、T1GGO/DONE ビットが自動的にクリアされます。ソフトウェアで T1GGO/DONE ビットを再度セットするまで、他のいかなるゲートイベントも Timer1 をインクリメントできません。詳細なタイミングは [図 22-5](#) を参照してください。

T1GCON レジスタの T1GSPM ビットをクリアしてシングルパルスゲートモードを無効にする場合、T1GGO/DONE ビットもクリアする必要があります。

トグルモードとシングルパルスモードを同時に有効にすると、両方の動作を併用できます。これにより、Timer1 ゲートソースのサイクル時間を計測できます。詳細なタイミングは [図 22-6](#) を参照してください。

22.6.5 Timer1 ゲート値ステータス

Timer1 ゲート値ステータスを使うと、ゲート制御値の最新のレベルを読み出す事ができます。この値は、T1GCON レジスタの T1GVAL ビットに格納されています。Timer1 ゲートが有効ではない (TMR1GE ビットがクリアされている) 場合でも T1GVAL ビットの値は有効です。

22.6.6 Timer1 ゲートイベント割り込み

Timer1 ゲートイベント割り込みが有効の場合、ゲートイベントの完了時に割り込みを生成できます。T1GVAL の立ち上がりエッジで、PIR1 レジスタの TMR1GIF フラグビットがセットされます。PIE1 レジスタの TMR1GIE ビットをセットすると、割り込みの生成が認識されます。

Timer1 ゲートが有効ではない (TMR1GE ビットがクリアされている) 場合でも、TMR1GIF フラグビットは動作します。

22.7 Timer1 割り込み

Timer1 のレジスタペア (TMR1H:TMR1L) は、FFFFh までインクリメントすると 0000h にロールオーバーします。Timer1 がロールオーバーすると、PIR1 レジスタの Timer1 割り込みフラグビットがセットされます。ロールオーバー割り込みを有効にするには、以下のビットをセットする必要があります。

- T1CON レジスタの ON ビット
- PIE1 レジスタの TMR1IE ビット
- INTCON レジスタの PEIE ビット
- INTCON レジスタの GIE ビット

割り込みサービスルーチンで TMR1IF ビットをクリアすると、割り込みは解除されます。

Note: 割り込みを有効にする前に、TMR1H:TMR1L レジスタペアと TMR1IF ビットをクリアする必要があります。

22.8 スリープ中の Timer1 の動作

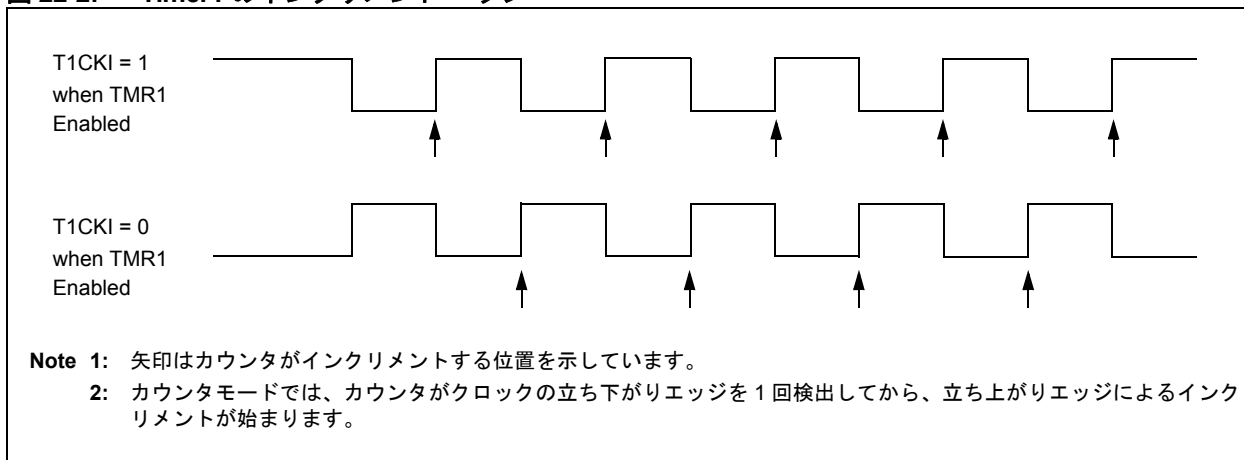
Timer1 は、非同期カウンタモードに設定されている場合にのみスリープ中に動作します。このモードでは、外部の水晶振動子またはクロック源によってカウンタをインクリメントします。Timer1 でデバイスを復帰させるには、以下の設定が必要です。

- T1CON レジスタの ON ビットをセットする
- PIE1 レジスタの TMR1IE ビットをセットする
- INTCON レジスタの PEIE ビットをセットする
- T1CON レジスタの SYNC ビットをセットする
- T1CON レジスタの CS ビットを設定する
- T1CON レジスタの OSCEN ビットを設定する

デバイスはオーバーフローによって復帰し、後続の命令を実行します。INTCON レジスタの GIE ビットがセットされている場合、デバイスは割り込みサービスルーチンを呼び出します。

セカンダリ オシレータは、 $\overline{\text{SYNC}}$ ビットの設定とは関係なくスリープ中でも動作を継続します。

図 22-2: Timer1 のインクリメント エッジ



22.9 CCP キャプチャ/コンペア タイムベース

CCP モジュールは、キャプチャまたはコンペアモードで動作中にタイムベースとして TMR1H:TMR1L レジスタペアを使います。

キャプチャモードでは、あらかじめ設定されたイベントの発生時に TMR1H:TMR1L レジスタペアの値が CCPR1H:CCPR1L レジスタペアにコピーされます。コンペアモードでは、CCPR1H:CCPR1L レジスタペアの値と TMR1H:TMR1L レジスタペアの値が一致するとイベントがトリガされます。このイベントは、自動変換トリガとして使えます。

詳細は [セクション 24.0 「キャプチャ/コンペア/PWM モジュール」](#) を参照してください。

22.10 CCP 自動変換トリガ

CCP が自動変換をトリガするように設定されている場合、トリガ発生時に TMR1H:TMR1L レジスタペアがクリアされます。この自動変換は、Timer1 割り込みを発生させません。この場合も、CCP モジュールは CCP 割り込みを生成するように設定できます。

この動作モードでは、CCPR1H:CCPR1L レジスタペアが Timer1 の周期レジスタとして機能します。

自動変換トリガを使うには、Timer1 を同期させ、クロック源として $F_{osc}/4$ を選択する必要があります。Timer1 を非同期モードで動作させると、自動変換トリガを検出できない場合があります。

TMR1H または TMR1L への書き込みと CCP からの自動変換トリガが競合した場合、書き込みが優先されます。詳細は [セクション 24.2.1 「自動変換トリガ」](#) を参照してください。

図 22-3: Timer1 ゲートイネーブル モード

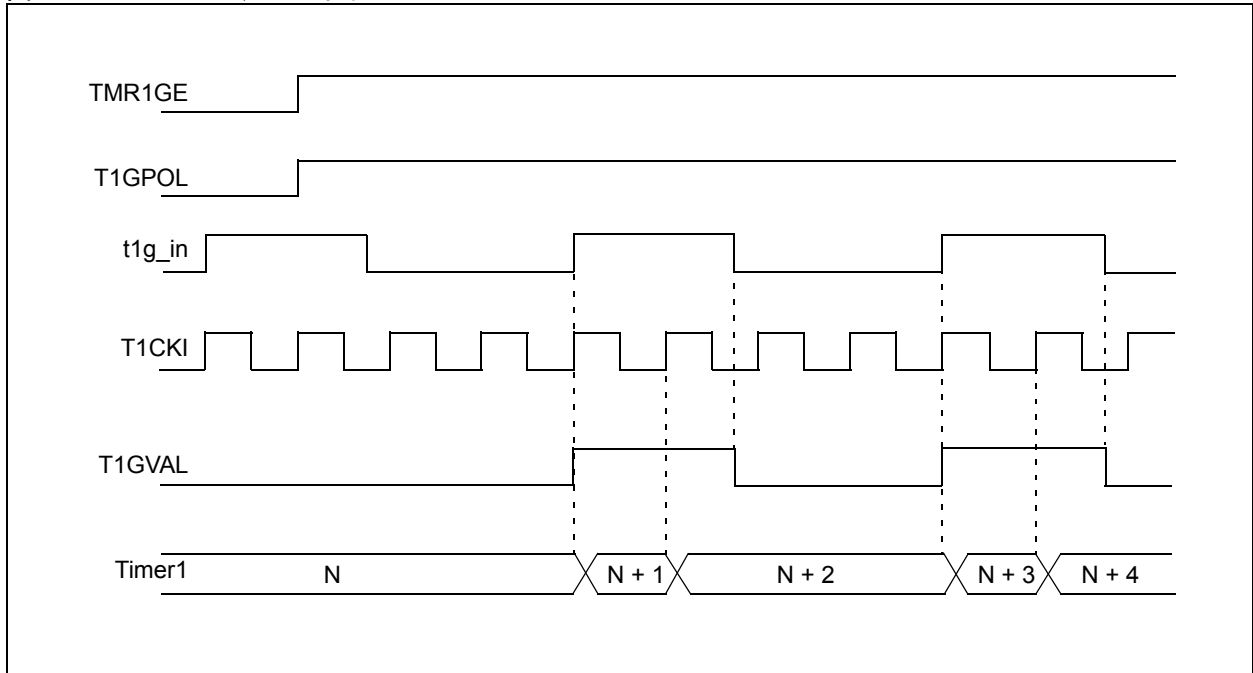
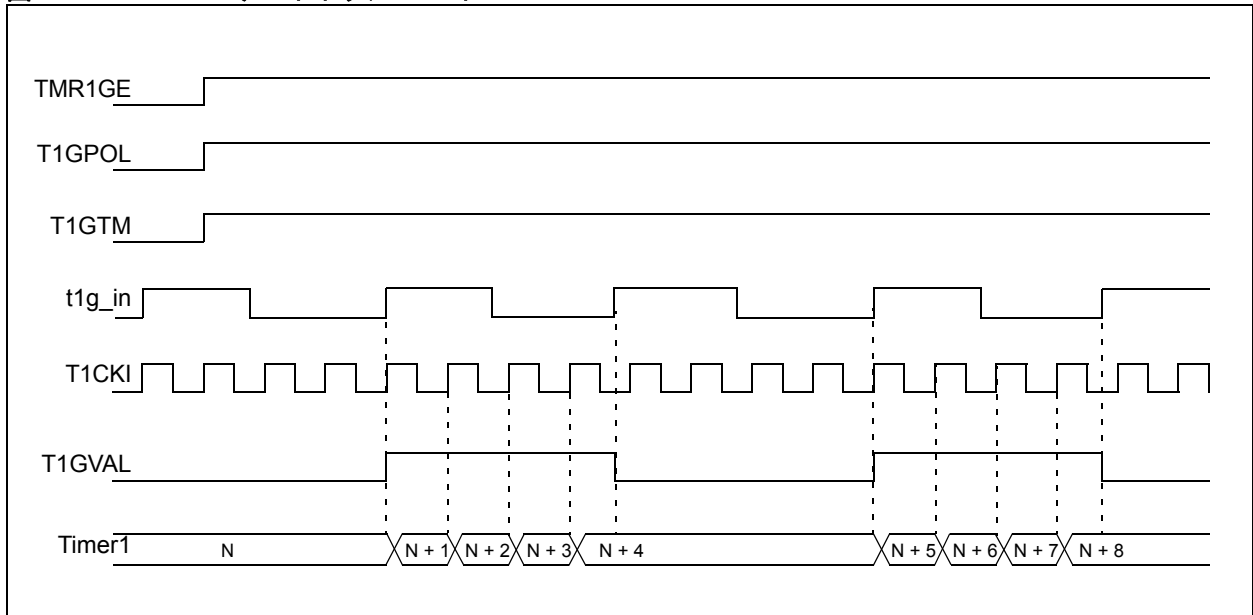


図 22-4: Timer1 ゲートトグル モード



PIC16(L)F1764/5/8/9

図 22-5: Timer1 ゲート シングルパルス モード

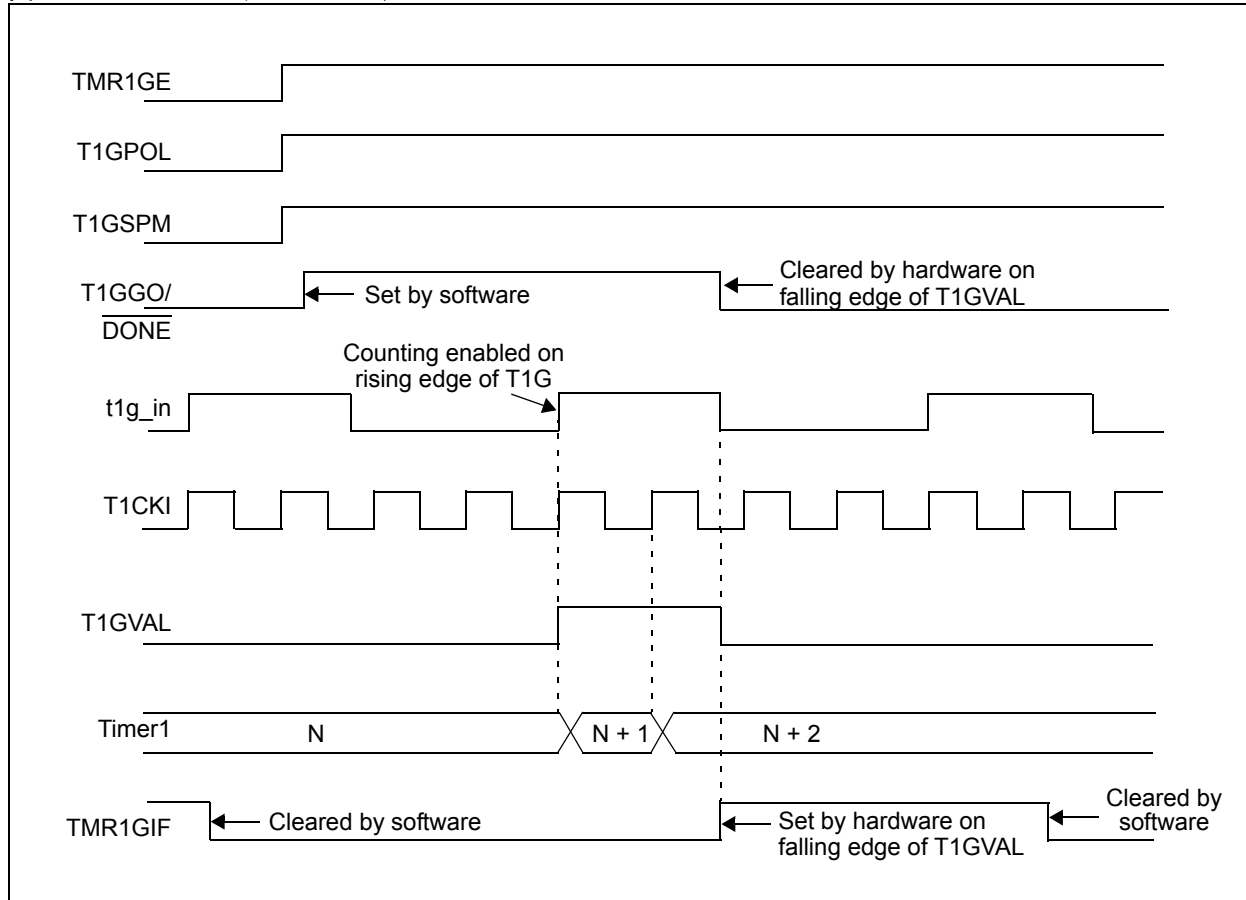
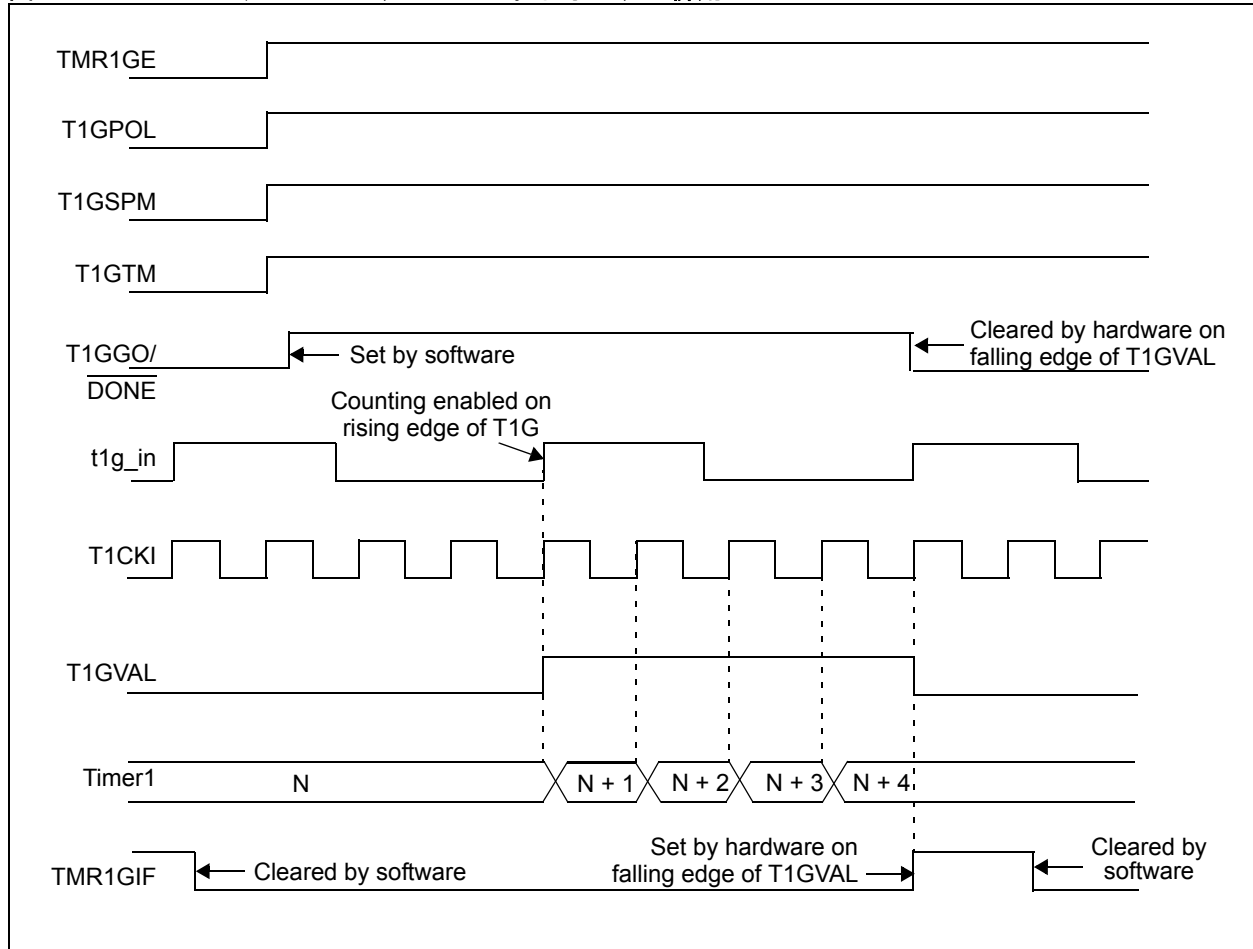


図 22-6: Timer1 ゲート シングルパルスおよびトグル併用モード



PIC16(L)F1764/5/8/9

22.11 レジスタ定義 :Timer1 制御

表 22-5 に Timer1 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1.2.2 「完全ビット名」](#) を参照してください。

表 22-5:

周辺モジュール	ビット名の接頭辞
Timer1	T1
Timer3	T3
Timer5	T5

レジスタ 22-1: T1CON: Timer1 制御レジスタ

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
CS<1:0>		CKPS<1:0>		OSCCN	$\overline{\text{SYNC}}$	—	ON
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-6 **CS<1:0>**: Timer1 クロック源選択ビット
11 = 予約済み (使用禁止)
10 = Timer1 クロック源にピンまたはオシレータを選択する :(1)
 T1OSCCN = 0 の場合:
 T1CKI ピンからの外部クロック (立ち上がりエッジ)
 T1OSCCN = 1 の場合:
 SOSCI/SOSCO ピンに接続された水晶振動子
01 = Timer1 クロック源にシステムクロック (Fosc) を選択する
00 = Timer1 クロック源に命令クロック (Fosc/4) を選択する
- bit 5-4 **CKPS<1:0>**: Timer1 入力クロック プリスケール選択ビット
11 = プリスケール値を 1:8 に設定する
10 = プリスケール値を 1:4 に設定する
01 = プリスケール値を 1:2 に設定する
00 = プリスケール値を 1:1 に設定する
- bit 3 **OSCCN**: LP オシレータ イネーブル制御ビット (1)
1 = 専用のセカンダリ オシレータ回路を有効にする
0 = 専用のセカンダリ オシレータ回路を無効にする
- bit 2 **SYNC**: Timer1 同期制御ビット
1 = 非同期クロック入力を同期しない
0 = 非同期クロック入力とシステムクロック (Fosc) を同期させる
- bit 1 **未実装**: 「0」として読み出し
- bit 0 **ON**: Timer1 ON ビット
1 = Timer1 を有効にする
0 = Timer1 を停止し Timer1 ゲート フリップフロップをクリアする

Note 1: Timer1 のみです。Timer3 と Timer5 では予約済みであり使えません。

レジスタ 22-2: T1GCON: Timer1 ゲート制御レジスタ

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
GE	GPOL	GTM	GSPM	GGO/ DONE	GVAL	GSS<1:0>	
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	HC = ビットはハードウェアでクリア

- bit 7 **GE:** Timer1 ゲート イネーブルビット
TMR1ON = 0 の場合:
 このビットを無視する
TMR1ON = 1 の場合:
 1 = Timer1 のカウントを Timer1 ゲート機能で制御する
 0 = Timer1 は、Timer1 ゲート機能とは無関係にカウントする
- bit 6 **GPOL:** Timer1 ゲート極性ビット
 1 = Timer1 ゲートはアクティブ High (ゲートが High の時にカウントする)
 0 = Timer1 ゲートはアクティブ Low (ゲートが Low の時にカウントする)
- bit 5 **GTM:** Timer1 ゲートトグル モードビット
 1 = Timer1 ゲートトグル モードを有効にする
 0 = Timer1 ゲートトグル モードを無効にし、トグル フリップフロップをクリアする
 Timer1 ゲート フリップフロップは立ち上がりエッジごとにトグルします。
- bit 4 **GSPM:** Timer1 ゲート シングルパルス モードビット
 1 = Timer1 ゲート シングルパルス モードを有効にし、Timer1 ゲートを制御する
 0 = Timer1 ゲート シングルパルス モードを無効にする
- bit 3 **GGO/DONE:** Timer1 ゲート シングルパルス アクイジション ステータスビット
 1 = Timer1 ゲート シングルパルス アクイジションがレディ状態でありエッジを待機している
 0 = Timer1 ゲート シングルパルス アクイジションは完了済み、または開始していない
- bit 2 **GVAL:** Timer1 ゲート値ステータスビット
 TMR1H:TMR1L へ提供される Timer1 ゲートの現在の状態を示します。
 Timer1 ゲートイネーブル (TMR1GE) の影響は受けません。
- bit 1-0 **GSS<1:0>:** Timer1 ゲートソース選択ビット
 11 = オプションで同期できるコンパレータ 2 出力 (sync_C2OUT)
 10 = オプションで同期できるコンパレータ 1 出力 (sync_C1OUT)
 01 = Timer0 オーバーフロー出力
 00 = Timer1 ゲートピン

PIC16(L)F1764/5/8/9

表 22-6: Timer1 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
CCPxCON	EN	OE	OUT	FMT	MODE<3:0>				259
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
TMRxH	16 ビット TMR1/3/5 レジスタの上位バイト保持レジスタ								220*
TMRxL	16 ビット TMR1/3/5 レジスタの下位バイト保持レジスタ								220*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TxCON	CS<1:0>		CKPS<1:0>		OSCCEN	SYNC	—	1ON	228
TxGCON	GE	GPOL	GTM	GSPM	GGO/ DONE	GVAL	GSS<1:0>		229

凡例: — = 未実装、「0」として読み出し。網掛けの部分は Timer1 モジュールでは使いません。

* 本文中でレジスタを説明しているページです。

Note 1: 未実装、「1」として読み出します。

2: PIC16(L)F1768/9 のみです。

23.0 Timer2/4/6 モジュール

Timer2/4/6 モジュールはワンショットおよび単安定モードで動作できる 8 ビットタイマであり、フリーランニング周期カウンタとしても、動作（スタート、実行、フリーズ、リセット）を制御する外部信号と組み合わせても動作できます。これらのタイマ動作とその他の内蔵周辺モジュール（コンパレータ、CCP モジュール等）を組み合わせると、パルス密度変調等の高度な波形制御が可能です。これらのタイマの特長は以下の通りです。

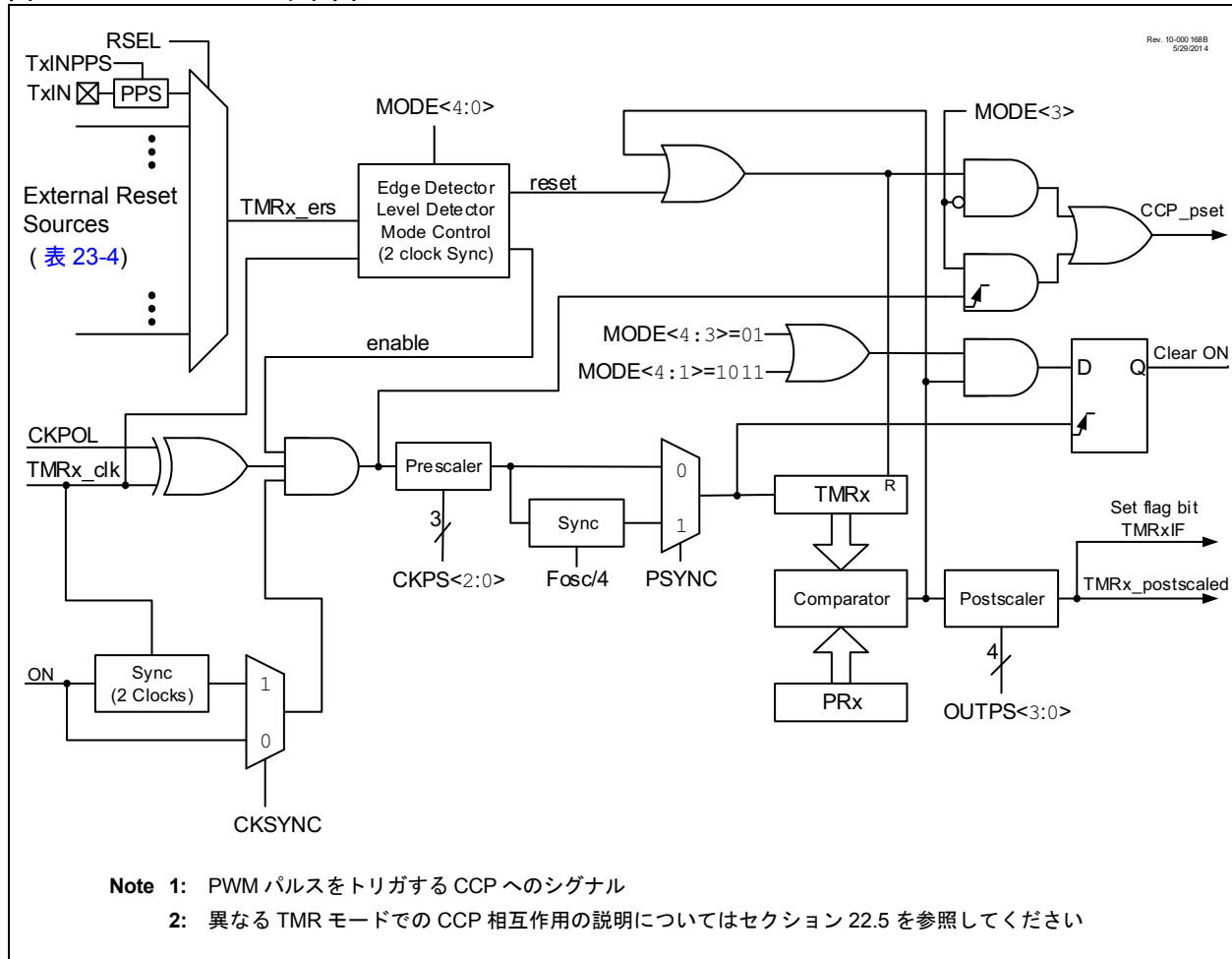
- 8 ビットのタイマレジスタ
- 8 ビットの周期レジスタ
- 選択可能な外部ハードウェア タイマリセット
- プログラマブルなプリスケアラ (1:1 ~ 1:128)
- プログラマブルなポストスケアラ (1:1 ~ 1:16)
- 同期 / 非同期動作を選択可能
- 代替クロック源
- 周期割り込み

- 3 種類の動作モード：
 - フリーランニング周期
 - ワンショット
 - 単安定

Timer2 のブロック図は図 23-1 を参照してください。クロック源のブロック図は図 23-2 を参照してください。

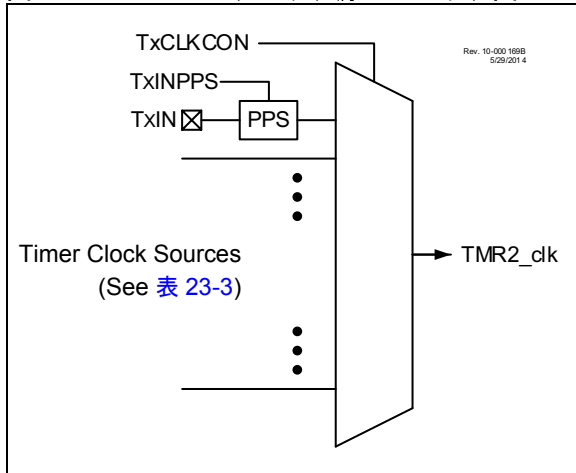
Note: このデバイスには Timer2 と同じモジュールがあと 2 つ実装されています。これらのタイマは Timer2、Timer4、Timer6 と呼ばれます。Timer2 に関する説明は Timer4 と Timer6 にも該当します。T2PR に関する説明は T4PR と T6PR にも該当します。

図 23-1: Timer2 ブロック図



PIC16(L)F1764/5/8/9

図 23-2: Timer2 クロック源のブロック図



23.1 Timer2 の動作

Timer2 の主な動作モード:

- フリーランニング周期
- ワンショット
- 単安定

各モードにはスタート、ストップ、リセットに関していくつかの選択肢があります。表 23-1 にその選択肢の一覧を示します。

全てのモードで、TMR2 カウントレジスタはプログラマブル プリスケアラからのクロック信号の立ち上がりエッジでインクリメントされます。TMR2 が T2PR と等しくなると、High レベルがポストスケアラ カウンタに出力されます。次のクロック入力で TMR2 がクリアされます。

ハードウェアからの外部信号でタイマ動作をゲート制御、または TMR2 カウントをリセットするように設定できます。ゲートモードでは、ゲートを OFF にするとカウンタは停止し ON にすると再開します。リセットモードでは、TMR2 カウントは外部信号源のレベルまたはエッジのどちらかでリセットされます。

TMR2 および T2PR レジスタは、どちらも直接読み書きできます。全ての任意のデバイスリセットで TMR2 レジスタはクリアされ、T2PR レジスタは FFh に初期化されます。以下のイベントでプリスケアラおよびポストスケアラ カウンタは両方共クリアされます。

- TMR2 レジスタへの書き込み
- T2CON レジスタへの書き込み
- 全てのデバイスリセット
- タイマをリセットする外部リセット信号源のイベント

Note: T2CON レジスタに書き込んでも TMR2 はクリアされません。

23.1.1 フリーランニング周期モード

クロックサイクルごとに TMR2 の値と周期レジスタ (T2PR) の値が比較されます。この 2 つの値が一致すると、コンパレータが次のサイクルで TMR2 の値を 00h にリセットし、出力ポストスケアラ カウンタをインクリ

メントします。ポストスケアラのカウンタが TMRxCON1 レジスタの OUTPS<4:0> ビットの値と等しくなると、TMR2_postscaled 出力に 1 クロック周期幅のパルスが発生し、ポストスケアラのカウンタがクリアされます。

23.1.2 ワンショットモード

ワンショット モードは基本的にフリーランニング周期モードと同じです。しかし TMR2 が T2PR と一致すると ON ビットがクリアされてタイマが停止し、T2ON ビットが OFF-ON と切り換わるまで再スタートしない点が異なります。タイマが最初の周期のイベントで停止し、タイマが再スタートした時にポストスケアラがリセットされるため、このモードではポストスケアラの「0」以外の OUTPS<4:0> 値は意味を持ちません。

23.1.3 単安定モード

単安定モードはワンショット モードと似ています。しかし ON ビットがクリアされず、タイマが外部リセットイベントで再スタートできる点が違いです。

23.2 Timer2 の出力

Timer2 モジュールの最も重要な出力は TMR2_postscaled であり、ポストスケアラ カウンタが TMR2xCON レジスタの OUTPS ビットの値と一致した時に TMR2_clk 周期 1 個分のパルスを出力します。T2PR ポストスケアラは、TMR2 値が T2PR 値と一致するごとにインクリメントされます。この信号は以下に示すいくつかの入力モジュールの入力としても選択できます。

- ADC モジュールの自動変換トリガ
- COG の自動シャットダウン要因

さらに、PWM モードにおけるパルス生成用の CCP モジュールも Timer2 を使います。TMR2 値はその他の内部信号と一緒に CCP モジュールに送られ、PWM 信号の周期とパルス幅の両方を適切に計測します。CCP と合わせて使う場合の Timer2 の設定の詳細は [セクション 24.6 「CCP/PWM クロックの選択」](#) を参照してください。また各種 Timer2 モードが CCP PWM 出力に与える影響の例は [セクション 23.5 「動作例」](#) を参照してください。

23.3 外部リセット信号源

Timer2 はクロック源の他に外部リセット信号源も使います。この外部リセット信号源は Timer2、Timer4、Timer6 に対して T2RST、T4RST、T6RST の各レジスタでそれぞれ選択します。この信号源を使うとタイマのモードに応じてタイマのスタート、ストップ、リセットを制御できます。タイマのモードは TMRxHLT レジスタの MODE<4:0> ビットで制御します。エッジトリガモードでは、外部トリガの間隔としてタイマクロック 6 周期が必要です。レベルトリガモードでは、少なくともタイマクロック 3 周期の長さのトリガレベルが必要です。デバッグフリーズ モード中は外部トリガを無視します。

表 23-1: Timer2 動作モード

モード	MODE<4:0>		出力動作	動作	タイマ制御		
	<4:3>	<2:0>			スタート	リセット	ストップ
フリーランニング 周期	00	000	周期パルス	ソフトウェアでゲート (図 23-4)	ON = 1	—	ON = 0
		001		ハードウェアでゲート (アクティブ High)(図 23-5)	ON = 1 かつ TMRx_ers = 1	—	ON = 0 または TMRx_ers = 0
		010		ハードウェアでゲート (アクティブ Low)	ON = 1 かつ TMRx_ers = 0	—	ON = 0 または TMRx_ers = 1
		011	周期パルス (ハードウェアリセット付き)	立ち上がりまたは立ち下がりエッジでリセット	ON = 1	TMRx_ers ↓	ON = 0
		100		立ち上がりエッジでリセット (図 23-6)		TMRx_ers ↑	
		101		立ち下がりエッジでリセット		TMRx_ers ↓	
		110		Low レベルでリセット		TMRx_ers = 0	ON = 0 または TMRx_ers = 0
		111		High レベルでリセット (図 23-7)		TMRx_ers = 1	ON = 0 または TMRx_ers = 1
ワンショット	01	000	ワンショット	ソフトウェアでスタート (図 23-8)	ON = 1	—	ON = 0 または TMRx = PRx の次の クロック (Note 2)
		001	エッジトリガスタート (Note 1)	立ち上がりエッジでスタート (図 23-9)	ON = 1 かつ TMRx_ers ↑	—	
		010		立ち下がりエッジでスタート	ON = 1 かつ TMRx_ers ↓	—	
		011		任意エッジでスタート	ON = 1 かつ TMRx_ers ↓	—	
		100	エッジトリガスタートとハードウェアリセット (Note 1)	立ち上がりエッジスタートと立ち上がりエッジでリセット (図 23-10)	ON = 1 かつ TMRx_ers ↑	TMRx_ers ↑	
		101		立ち下がりエッジスタートと立ち下がりエッジでリセット	ON = 1 かつ TMRx_ers ↓	TMRx_ers ↓	
		110		立ち上がりエッジスタートと Low レベルリセット (図 23-11)	ON = 1 かつ TMRx_ers ↑	TMRx_ers = 0	
		111		立ち下がりエッジスタートと High レベルリセット	ON = 1 かつ TMRx_ers ↓	TMRx_ers = 1	
単安定	10	000	予約済み				
		001	エッジトリガスタート (Note 1)	立ち上がりエッジでスタート (図 23-12)	ON = 1 かつ TMRx_ers ↑	—	ON = 0 または TMRx = PRx の次の クロック (Note 3)
		010		立ち下がりエッジでスタート	ON = 1 かつ TMRx_ers ↓	—	
		011		任意エッジでスタート	ON = 1 かつ TMRx_ers ↓	—	
予約済み	100	予約済み					
予約済み	101	予約済み					
ワンショット	10	110	レベルトリガスタートとハードウェアリセット	High レベルスタートと Low レベルリセット (図 23-13)	ON = 1 かつ TMRx_ers = 1	TMRx_ers = 0	ON = 0 またはリセット時も維持 (Note 2)
		111		Low レベルスタートと High レベルリセット	ON = 1 かつ TMRx_ers = 0	TMRx_ers = 1	
予約済み	11	xxx	予約済み				

- Note 1:** ON = 0 になると、次に ON = 1 になった後にタイマを再スタートするにはエッジが 1 個必要です。
Note 2: TMRx = PRx になると、次のクロックで ON はクリアされ TMRx は 00h で停止します。
Note 3: TMRx = PRx になると、次のクロックで TMRx は 00h で停止しますが ON はクリアされません。

PIC16(L)F1764/5/8/9

23.4 Timer2 割り込み

Timer2は、デバイス割り込みを生成する事もできます。この割り込みは、ポストスケアラ カウンタが16通りのポストスケール設定(1:1 ~ 1:16)の1つと一致した時に生成されます。このポストスケール設定値はT2CONレジスタのポストスケアラ制御ビットOUTPS<3:0>で選択します。PIE1レジスタのTMR2IE割り込みイネーブルビットをセットすると、割り込みが有効になります。図23-3に割り込みのタイミングを示します。

図 23-3: Timer2 プリスケアラ、ポストスケアラ、割り込みのタイミング図



23.5 動作例

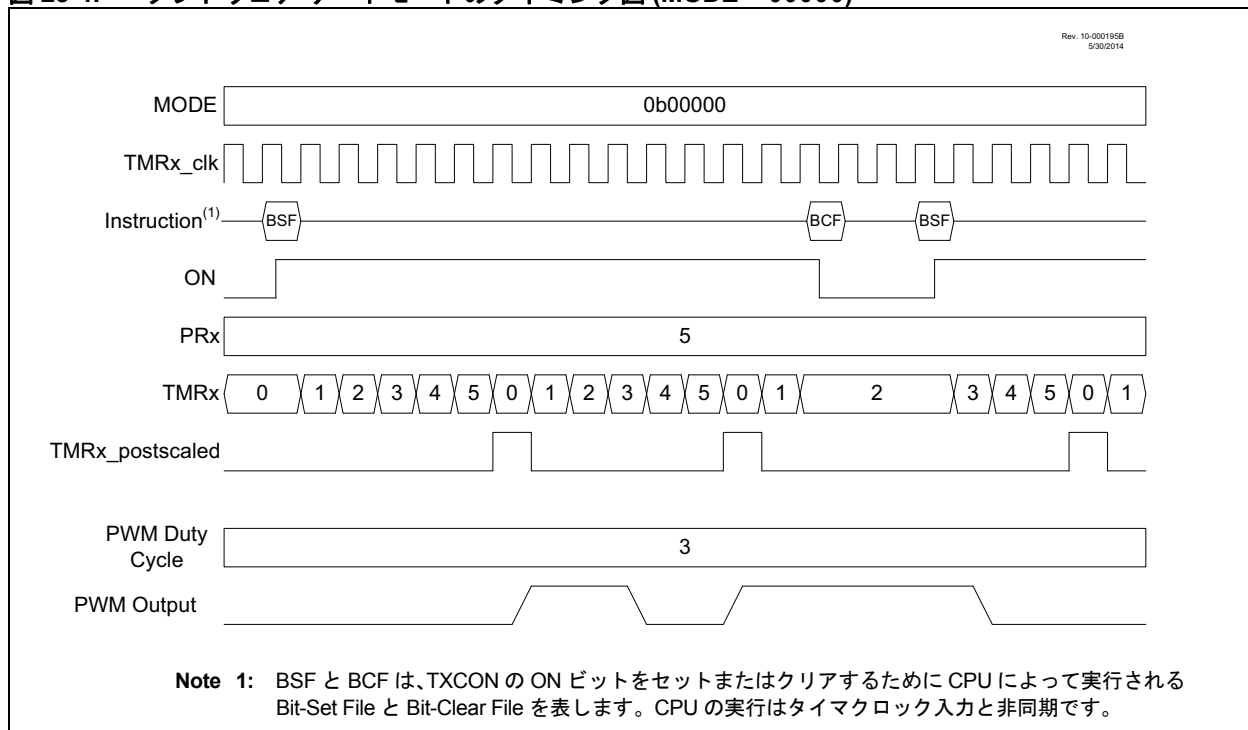
特に明記しない場合、以下の注釈を以下のタイミング図に適用します。

- プリスケアラとポストスケアラの両方を 1:1 に設定します (TxCONレジスタのCKPSおよびOUTPSビットを両方クリア)。
- タイミング図では $F_{osc}/4$ を除く全クロックを表示し、ON と Timer2_ers の両方について少なくとも完全な2周期分のクロック同期遅延を示しています。 $F_{osc}/4$ を使う場合、Timer2_ers のクロック同期遅延は少なくとも 1 命令周期です (次の命令周期で ON が印加されます)。
- PWM デューティ サイクルと PWM 出力は、[セクション 24.6 「CCP/PWM クロックの選択」](#) で説明したように CCP モジュールの PWM 機能にタイマを使うものとして示しています。これらの信号は Timer2 モジュールの一部ではありません。

23.5.1 ソフトウェア ゲートモード

このモードはレガシー Timer2 動作に対応します。タイマは $ON = 1$ の時クロック入力ごとにインクリメントし、 $ON = 0$ の時はインクリメントしません。TMRx カウントがPRx周期カウントと等しくなるとタイマは次のクロックでリセットされ、0 からカウントを続けます。[図 23-4](#) に、ON ビットをソフトウェアで制御する場合の動作を示します。PRx = 5 の場合、カウンタは TMRx = 5 まで進むと次のクロックでゼロになります。

図 23-4: ソフトウェア ゲートモードのタイミング図 (MODE = 00000)



PIC16(L)F1764/5/8/9

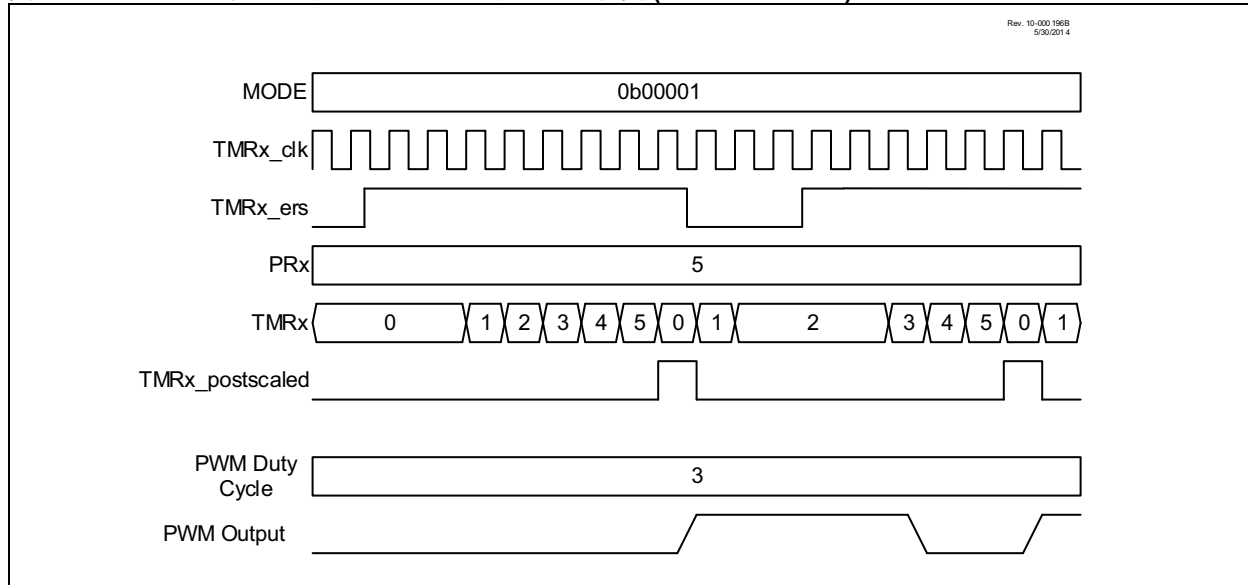
23.5.2 ハードウェア ゲートモード

ハードウェア ゲートモードはソフトウェア ゲートモードと同様に動作しますが、TMRx_ers 外部信号でもタイマをゲート制御できる点が異なります。CCP と一緒に使うと、ゲート制御で PWM 周期を延長できます。PWM 出力が High の時にタイマが停止すると、デューティ サイクルも延長されます。

MODE<4:0> = 00001 の場合、タイマは外部信号が High になると停止します。MODE<4:0> = 00010 の場合、タイマは外部信号が Low になると停止します。

図 23-5 に MODE<4:0> = 00001 (High 入力レベルでカウンタがスタート) の場合のハードウェア ゲートモードを示します。

図 23-5: ハードウェア ゲートモードのタイミング図 (MODE = 00001)



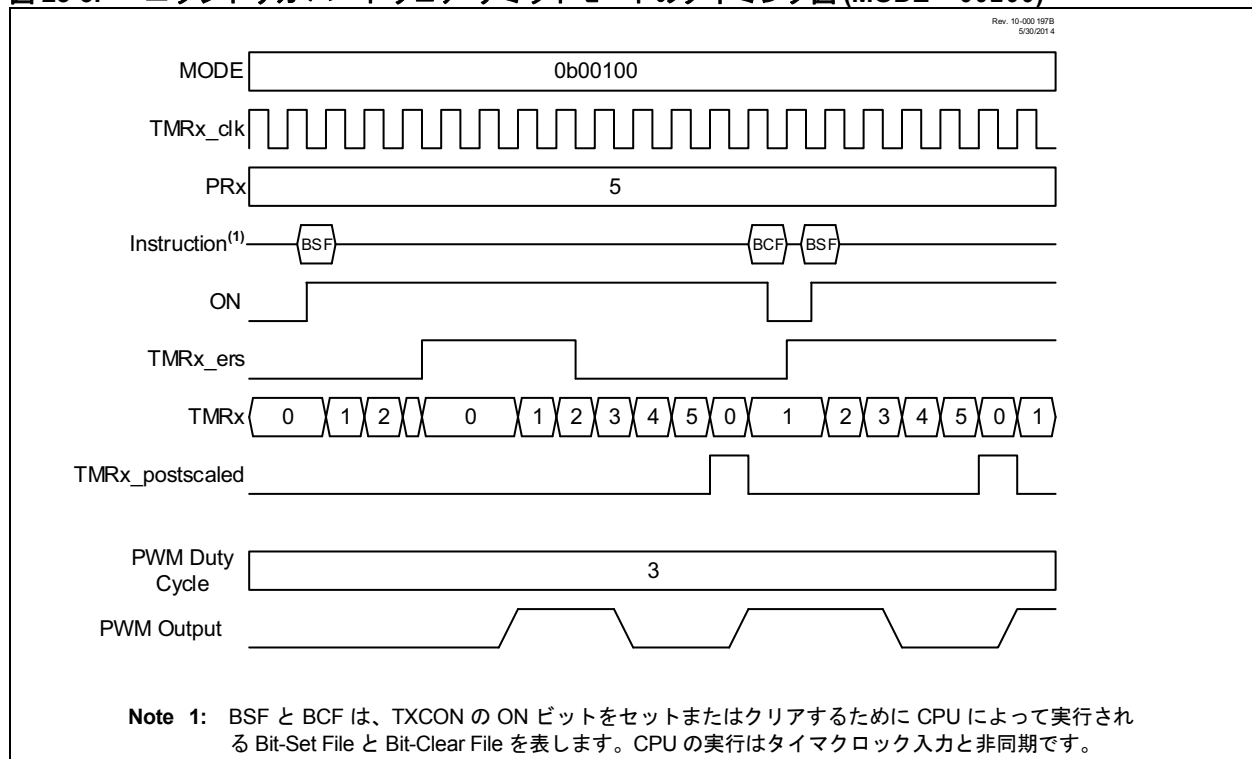
23.5.3 エッジトリガハードウェアリミットモード

ハードウェアリミットモードでは、タイマが周期カウントに達する前に TMRx_ers 外部信号でタイマをリセットできます。以下 3 種類のリセットを使えます。

- 立ち上がりまたは立ち下がりエッジでのリセット (MODE<4:0> = 00011)
- 立ち上がりエッジでのリセット (MODE<4:0> = 00100)
- 立ち下がりエッジでのリセット (MODE<4:0> = 00101)

タイマを PWM モードで CCP と一緒に使う場合、周期途中のリセットで周期を短縮し、2 クロック遅延の後 PWM パルスを再開します。図 23-6 を参照してください。

図 23-6: エッジトリガハードウェアリミットモードのタイミング図 (MODE = 00100)



PIC16(L)F1764/5/8/9

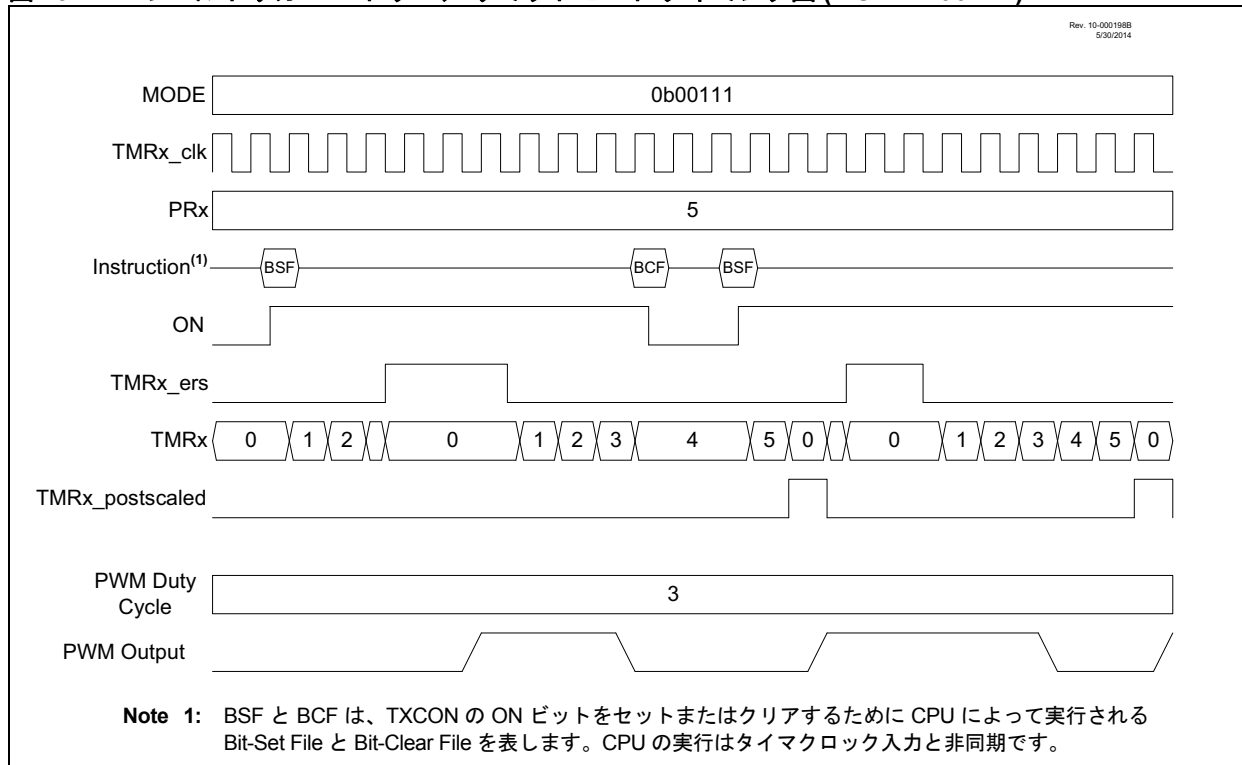
23.5.4 レベルトリガハードウェアリミットモード

図 23-7 に示すように、レベルトリガハードウェアリミットタイマモードでは、TMRx_ers 外部信号の High または Low レベルでカウンタがリセットされます。MODE<4:0> = 00110 を選択すると外部信号が Low レベルの時にタイマをリセットします。MODE<4:0> = 00111 を選択すると外部信号が High レベルの時にタイマをリセットします。例えば、TMRx_ers = 1 の期間中タイマはリセットされています。ON は BSF および BCF 命令で制御されます。ON = 0 の場合、外部信号は無視されます。

CCP が PWM のタイムベースとしてタイマを使う場合、PWM 出力はタイマのカウンタ開始時 High に設定され、タイマカウンタが CCPRx の値と一致した時のみ Low に設定されます。タイマカウンタが PRx の値と一致した時と、外部リセット信号が真のまま 2 クロック周期経過した時のどちらかで、タイマはリセットされます。

PRx 一致直後のクロックと、外部リセット信号がリセットを解除して 2 クロック周期経過した時のどちらかで、タイマはカウントを開始し PWM 出力は High に設定されます。タイマが CCPRx パルス幅値と一致するまでカウントする間、PWM 出力は High を維持します。PWM 出力が High の間に外部リセット信号が真になった場合、リセット信号が解放されるまで PWM 出力は High に維持され、タイマは CCPRx 値と一致するまでカウントできます。

図 23-7: レベルトリガハードウェアリミットモードタイミング図 (MODE = 00111)

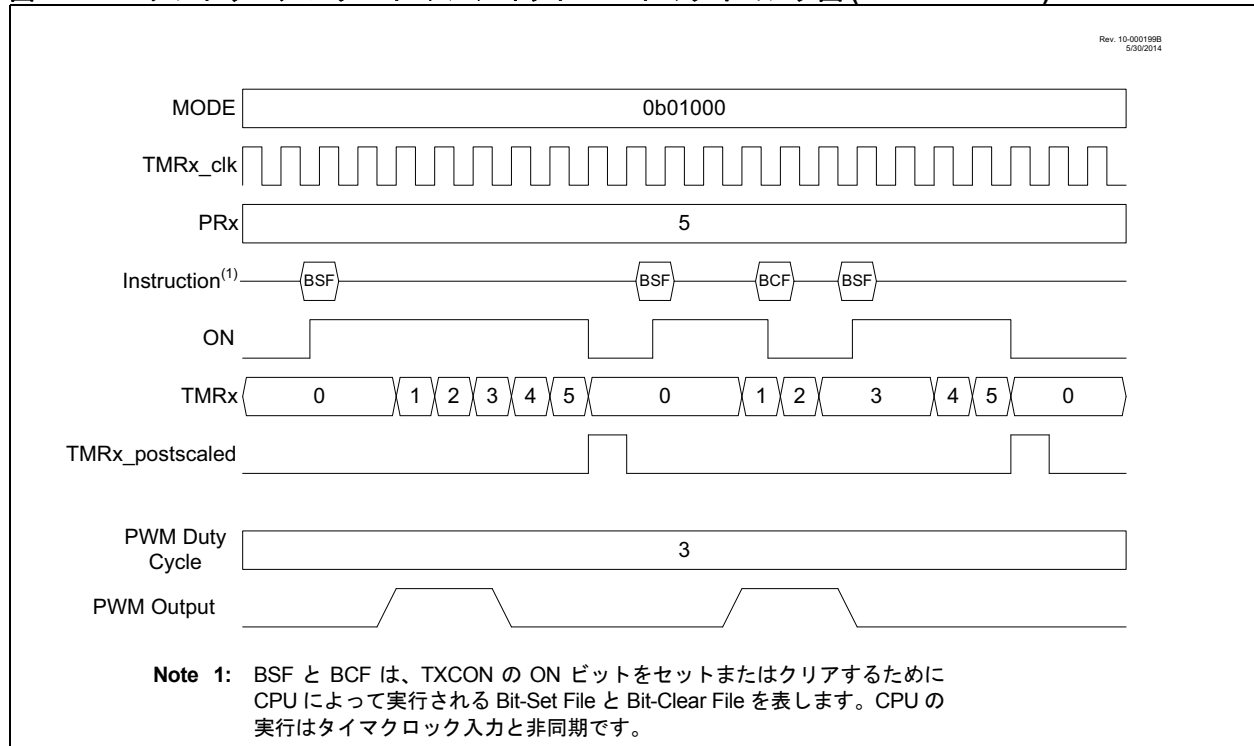


23.5.5 ソフトウェア スタート ワンショット モード

ワンショットモードでは、タイマ値が PRx 周期値に一致した時タイマはリセットされ ON ビットはクリアされます。次のタイマサイクルを開始するにはソフトウェアで ON ビットをセットする必要があります。図 23-8 に示すワンショットモードを選択するには、MODE<4:0> = 01000 に設定します。以下の例では ON は BSF および BCF 命令で制御します。前者では、BSF 命令で ON をセットし、最後までカウントしたら ON をクリアしています。後者では、BSF 命令でサイクルを開始し、BCF/BSF 命令でサイクル中にカウンタを OFF/ON した後、最後までカウントしています。

CCP PWM動作と合わせてワンショットモードを使う場合、PWM パルス駆動は ON ビットのセットと同時に開始します。PWM 駆動がアクティブな状態で ON ビットをクリアすると PWM 駆動は延長されます。PWM 駆動はタイマ値が CCPRx パルス幅値に一致した時点で終了します。ソフトウェアで ON ビットをセットして次のサイクルを開始するまで、PWM駆動は OFF に維持されます。CCPRx 一致後で PRx 一致前にソフトウェアが ON ビットをクリアする場合、ON ビットがクリアされている時間だけ PWM 駆動が延長されます。PRx周期カウント一致で ON ビットがクリアされた後、次のタイミングサイクルを開始するには ON ビットをセットする必要があります。

図 23-8: ソフトウェア スタート ワンショット モードのタイミング図 (MODE = 01000)



PIC16(L)F1764/5/8/9

23.5.6 エッジトリガ ワンショット モード

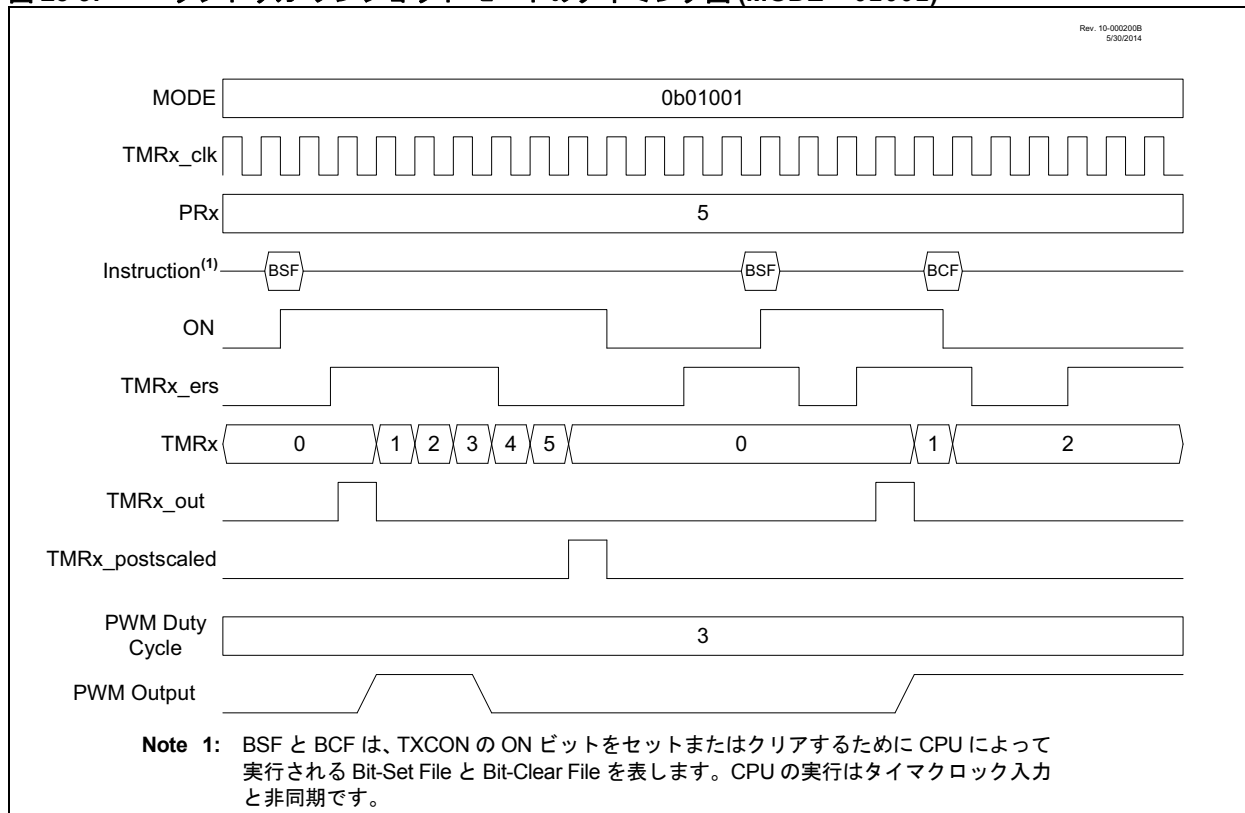
エッジトリガ ワンショット モードでは、ON ビットをセットした後外部信号入力のエッジでタイマをスタートし、タイマが PRx 周期値と一致した時 ON ビットをクリアします。以下のエッジでタイマをスタートします。

- 立ち上がりエッジ (MODE<4:0> = 01001)
- 立ち下がりエッジ (MODE<4:0> = 01010)
- 立ち上がりまたは立ち下がりエッジ (MODE<4:0> = 01011)

ON ビットがクリアされタイマが停止した場合、カウントを再開するには ON ビットをセットした後、次の TMRx_ers エッジが必要です。図 23-9 に、立ち上がりエッジ ワンショット モードの動作を示します。

エッジトリガ ワンショット モードを CCP と併用する場合、エッジトリガが PWM 駆動を ON します。またタイマが CCPRx パルス幅値と一致した時 PWM 駆動を OFF し、PRx 周期カウンタ一致でタイマが停止している間 OFF を維持します。

図 23-9: エッジトリガワンショットモードのタイミング図 (MODE = 01001)



23.5.7 エッジトリガハードウェアリミットワンショットモード

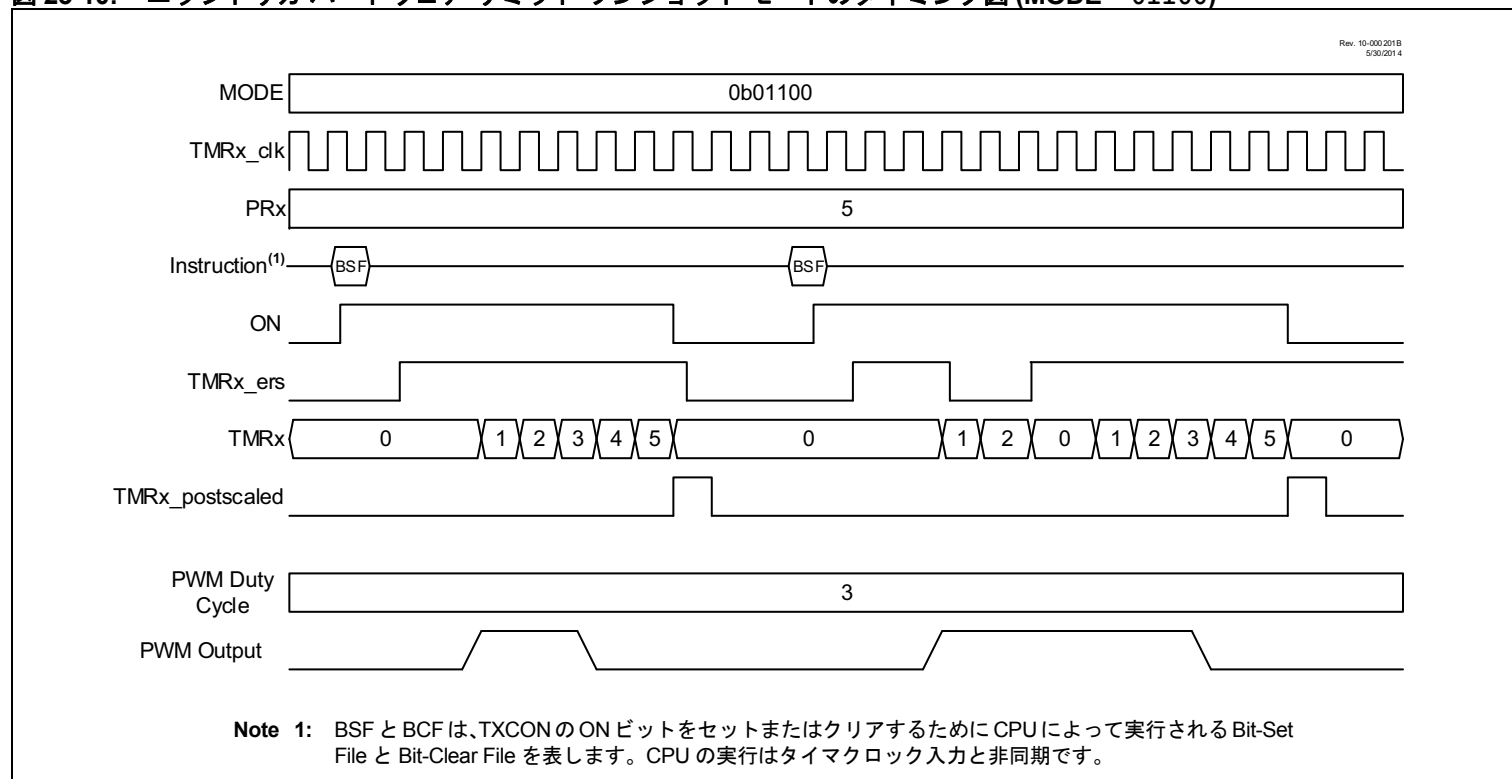
エッジトリガハードウェアリミットワンショットモードでは、ON ビットをセットした後外部信号の最初のエッジでタイマをスタートし、その後の全てのエッジでリセットします。タイマをスタートするために必要なものは、ON ビットをセットした後の最初のエッジだけです。カウンタはその後の全ての外部リセットエッジから 2 クロック後に自動的にカウントを再開します。エッジトリガを以下に示します。

- 立ち上がりエッジスタートおよびリセット (MODE<4:0> = 01100)
- 立ち下がりエッジスタートおよびリセット (MODE<4:0> = 01101)

タイマ値が PRx 周期値に一致した時、タイマはリセットし ON ビットをクリアします。ソフトウェアが ON ビットをセットしない限り外部信号エッジは影響を与えません。図 23-10 に、立ち上がりエッジハードウェアリミットワンショット動作を示します。

このモードを CCP と一緒に使う場合、最初のエッジトリガとその後の全てのリセットエッジで PWM 駆動が ON になります。外部信号エッジが一致前にタイマをリセットしない限り、PWM 駆動はタイマが CCPRx パルス幅値と一致した時 OFF になり、PRx 周期カウンタ一致でタイマが停止している間 OFF を維持します。

図 23-10: エッジトリガハードウェアリミットワンショットモードのタイミング図 (MODE = 01100)



23.5.8 レベルリセット、エッジトリガハードウェアリミットワンショットモード

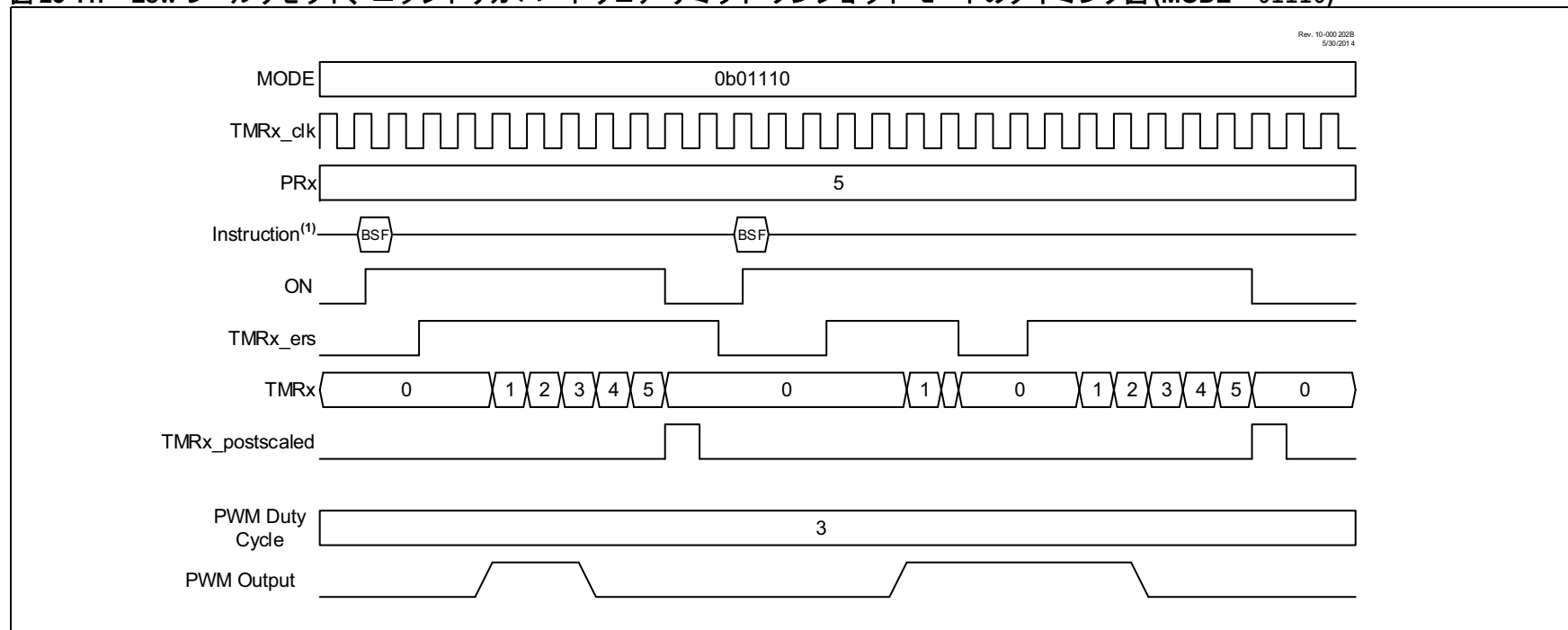
レベルトリガワンショットモードでは、タイマカウントは外部信号レベルでリセットされ、ON ビットがセットされている間はリセットレベルからアクティブレベルへの遷移の立ち上がり/立ち下がりエッジでカウントを開始します。リセットレベルは以下のように選択されます。

- Low リセットレベル (MODE<4:0> = 01110)
- High リセットレベル (MODE<4:0> = 01111)

タイマカウントがPRx周期カウントと一致すると、タイマはリセットされONビットはクリアされます。PRx一致とソフトウェア制御のどちらかでONビットがクリアされると、カウンタをスタートするには、ONビットのセット後に次の外部信号エッジが必要です。

レベルトリガリセットワンショットモードをCCP PWM動作と一緒に使う場合、タイマをスタートさせる外部信号エッジでPWM駆動がONになります。PWM駆動は、タイマ値がCCPRxパルス幅値に一致した時にOFFになります。PRx周期一致でタイマカウントがクリアされてもPWM駆動はONになりません。

図 23-11: Low レベルリセット、エッジトリガハードウェアリミットワンショットモードのタイミング図 (MODE = 01110)



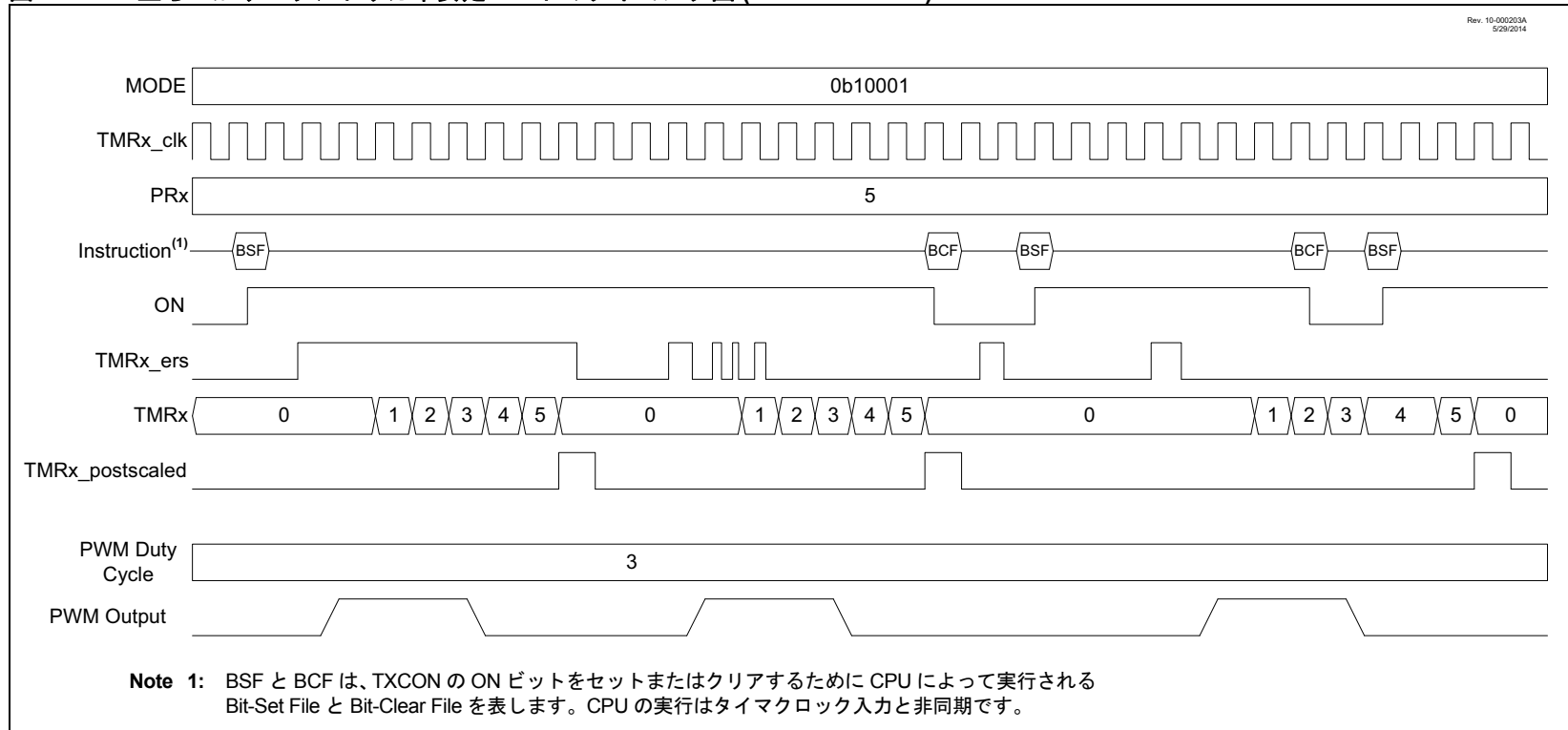
23.5.9 エッジトリガ単安定モード

エッジトリガ単安定モードでは、ON ビットをセットした後外部リセット信号入力のエッジでタイマをスタートし、タイマが PRx 周期値と一致した時インクリメントを停止します。以下のエッジでタイマをスタートします。

- 立ち上がりエッジ (MODE<4:0> = 10001)
- 立ち下がりエッジ (MODE<4:0> = 10010)
- 立ち上がりまたは立ち下がりエッジ (MODE<4:0> = 10011)

エッジトリガ単安定モードを CCP PWM 動作と一緒に使う場合、タイマをスタートさせる外部リセット信号エッジで PWM 駆動が ON になりますが、タイマが PRx 値と一致しても ON になりません。タイマがインクリメントしている間、外部リセット信号にエッジが生じても CCP PWMI には影響を与えません。

図 23-12: 立ち上がりエッジトリガ単安定モードのタイミング図 (MODE = 10001)



23.5.10 レベルトリガハードウェアリミットワンショットモード

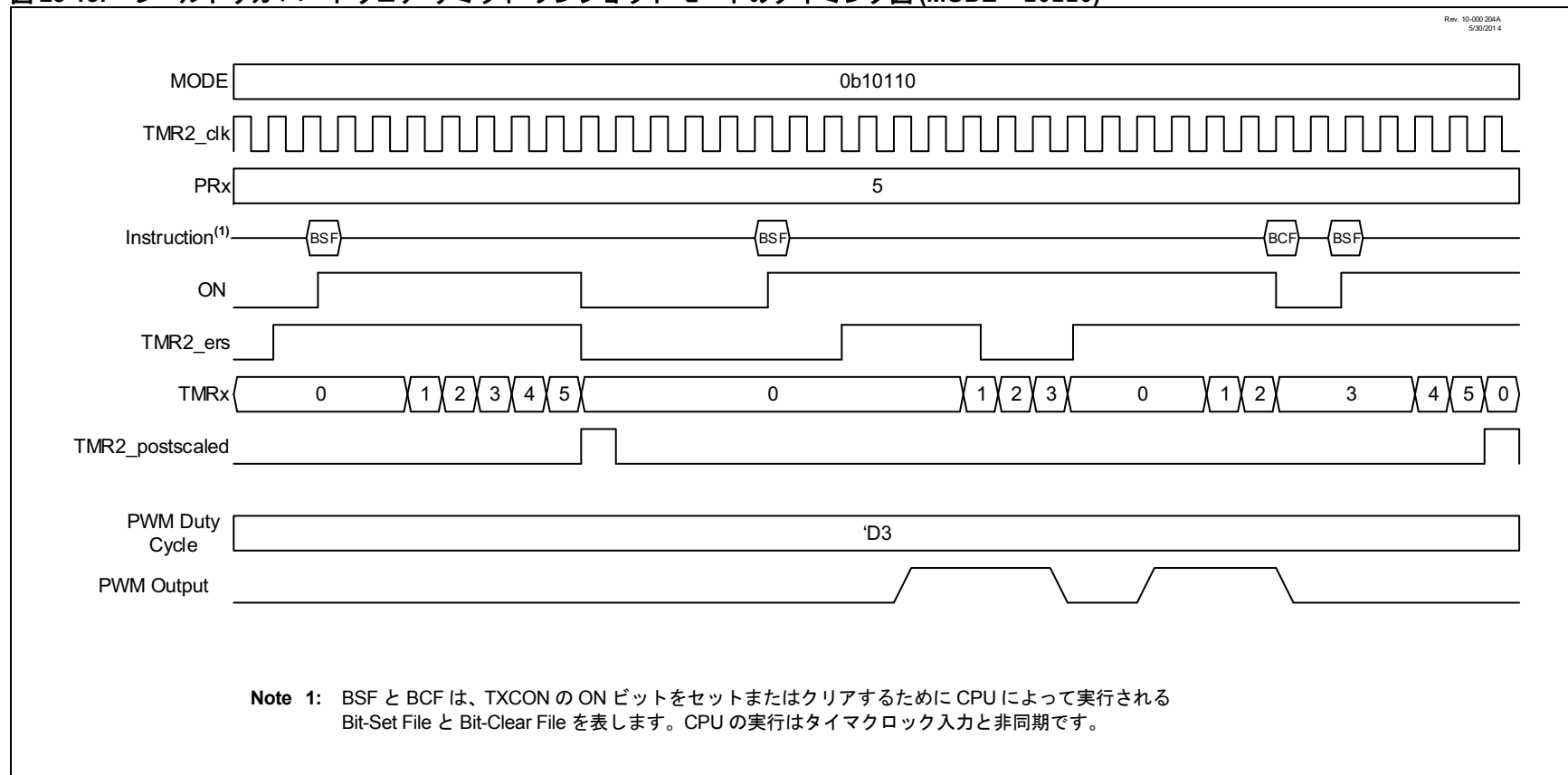
レベルトリガハードウェアリミットワンショットモードでは、外部信号がリセットレベルの時タイマをリセット状態に保持し、ONビットがセットされかつ外部信号がリセットレベルでない時にカウントを開始します。外部信号がリセットレベルでないかONビットがセットされているかどちらかの条件が満たされている場合、もう一方の条件が満たされるとタイマはスタートします。リセットレベルは以下のように選択されます。

- Low リセットレベル (MODE<4:0> = 10110)
- High リセットレベル (MODE<4:0> = 10111)

タイマカウントがPRx周期カウントと一致すると、タイマはリセットされONビットはクリアされます。PRx一致とソフトウェア制御のどちらかでONビットがクリアされると、ONビットがセットされかつ外部信号がリセットレベルでなくなるまでタイマはリセット状態を維持します。

レベルトリガハードウェアリミットワンショットモードをCCP PWM動作と一緒に使う場合、外部信号エッジとONビットのセットのどちらかでPWM駆動はONし、タイマをスタートさせます。

図 23-13: レベルトリガハードウェアリミットワンショットモードのタイミング図 (MODE = 10110)



23.6 スリープ中の Timer2 の動作

PSYNC = 1 の場合、プロセッサがスリープ中は Timer2 を動作させる事はできません。プロセッサがスリープ中は、TMR2 および T2PR レジスタの内容は変更されません。

PSYNC = 0 の場合、Timer2 は選択したクロック源が動作している限りスリープ中も動作します。LFINTOSC、MFINTOSC、HFINTOSC の各オシレータのいずれかをタイマクロック源として選択すると、そのオシレータはスリープ中も動作を継続します。

PIC16(L)F1764/5/8/9

23.7 レジスタ定義 : Timer2/4/6 制御

表 23-2 に Timer2/4/6 周辺モジュールの完全ビット名の接頭辞を示します。詳細は[セクション 1.1.2.2 「完全ビット名」](#)を参照してください。

表 23-2:

周辺モジュール	ビット名の接頭辞
Timer2	T2
Timer4	T4
Timer6	T6

レジスタ 23-1: TxCLKCON: Timerx クロック選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CS<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **CS<3:0>:** Timerx クロック選択ビット
[表 23-3](#) を参照してください。

表 23-3: Timerx のクロック源

CS<3:0>	Timer2	Timer4	Timer6
1011-1111	予約済み	予約済み	予約済み
1010	LC3_out	LC3_out	LC3_out
1001	LC2_out	LC2_out	LC2_out
1000	LC1_out	LC1_out	LC1_out
0111	ZCD_out	ZCD_out	ZCD_out
0110	SOSC	SOSC	SOSC
0101	MFINTOSC	MFINTOSC	MFINTOSC
0100	LFINTOSC	LFINTOSC	LFINTOSC
0011	HFINTOSC	HFINTOSC	HFINTOSC
0010	Fosc	Fosc	Fosc
0001	Fosc/4	Fosc/4	Fosc/4
0000	T2INPPS で選択されたピン	T4INPPS で選択されたピン	T6INPPS で選択されたピン

レジスタ 23-2: TxCON: Timerx 制御レジスタ

R/W/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ON ⁽¹⁾	CKPS<2:0>			OUTPS<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	HC = ビットはハードウェアでクリア

bit 7 **ON:** Timerx ON ビット
 1 = Timerx を ON にする
 0 = Timerx を OFF にする。全てのカウンタおよびステートマシンをリセットする

bit 6-4 **CKPS<2:0>:** Timer2 型クロック プリスケーラ選択ビット
 111 = 1:128
 110 = 1:64
 101 = 1:32
 100 = 1:16
 011 = 1:8
 010 = 1:4
 001 = 1:2
 000 = 1:1

bit 3-0 **OUTPS<3:0>:** Timerx 出力ポストスケーラ選択ビット
 1111 = 1:16
 1110 = 1:15
 1101 = 1:14
 1100 = 1:13
 1011 = 1:12
 1010 = 1:11
 1001 = 1:10
 1000 = 1:9
 0111 = 1:8
 0110 = 1:7
 0101 = 1:6
 0100 = 1:5
 0011 = 1:4
 0010 = 1:3
 0001 = 1:2
 0000 = 1:1

Note 1: 一部のモードでは、ON ビットはハードウェアで自動的にクリアされます。[セクション 23.5「動作例」](#)を参照してください。

PIC16(L)F1764/5/8/9

レジスタ 23-3: TxHLT: Timerx ハードウェア リミット制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PSYNC ^(1, 2)	CKPOL ⁽³⁾	CKSYNC ^(4, 5)	MODE<4:0> ^(6, 7)				
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7 **PSYNC:** Timerx プリスケアラ同期イネーブルビット (1, 2)
1 = TMRx プリスケアラ出力を Fosc/4 に同期させる
0 = TMRx プリスケアラ出力を Fosc/4 に同期させない
- bit 6 **CKPOL:** Timerx クロック極性選択ビット (3)
1 = 入力クロックの立ち下がりがエッジでタイマ / プリスケアラをクロッキングする
0 = 入力クロックの立ち上がりがエッジでタイマ / プリスケアラをクロッキングする
- bit 5 **CKSYNC:** Timerx クロック同期イネーブルビット (4, 5)
1 = ON レジスタビットを TMR2_clk 入力に同期させる
0 = ON レジスタビットを TMR2_clk 入力に同期させない
- bit 4-0 **MODE<4:0>:** Timerx 制御モード選択ビット (6, 7)
[表 23-1](#) を参照してください。

- Note 1:** このビットをセットしておく、TMRx を読み出す時確実に有効な値を返します。
- 2:** このビットが「1」の場合、Timer2 はスリープ中は動作できません。
- 3:** ON = 1 の間、CKPOL は変更できません。
- 4:** このビットをセットしておく、ON が有効でも無効でもグリッチのない動作が保証されます。
- 5:** ON ビットをセットした後、タイマ動作は TMRx 入力 2 クロック分遅延します。
- 6:** 特に明記しない限り全てのモードで ON = 1 でスタートし、ON = 0 でストップします (ストップは TMRx 値に影響しません)。
- 7:** TMRx = PRx になると、動作モードに関わらず次のクロックで TMRx がクリアされます。

レジスタ 23-4: TxRST: Timerx 外部リセット信号選択レジスタ

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	RSEL<4:0>				
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-5 **未実装:** 「0」として読み出し
bit 4-0 **RSEL<4:0>:** Timerx 外部リセット信号源選択ビット
 [表 23-4](#) を参照してください。

表 23-4: 外部リセット信号源

RSEL<4:0>	Timer2	Timer4	Timer6
10010-11111	予約済み	予約済み	予約済み
10001	LC3_out	LC3_out	LC3_out
10000	LC2_out	LC2_out	LC2_out
01111	LC1_out	LC1_out	LC1_out
01110	ZCD_out	ZCD_out	ZCD_out
01101	sync_C4OUT ⁽¹⁾	sync_C4OUT ⁽¹⁾	sync_C4OUT ⁽¹⁾
01100	sync_C3OUT ⁽¹⁾	sync_C3OUT ⁽¹⁾	sync_C3OUT ⁽¹⁾
01011	sync_C2OUT	sync_C2OUT	sync_C2OUT
01010	sync_C1OUT	sync_C1OUT	sync_C1OUT
01001	PWM6_out ⁽¹⁾	PWM6_out ⁽¹⁾	PWM6_out ⁽¹⁾
01000	PWM5_out	PWM5_out	PWM5_out
00111	PWM4_out ⁽¹⁾	PWM4_out ⁽¹⁾	PWM4_out ⁽¹⁾
00110	PWM3_out	PWM3_out	PWM3_out
00101	CCP2_out ⁽¹⁾	CCP2_out ⁽¹⁾	CCP2_out ⁽¹⁾
00100	CCP1_out	CCP1_out	CCP1_out
00011	TMR6_postscaled	TMR6_postscaled	予約済み
00010	TMR4_postscaled	予約済み	TMR4_postscaled
00001	予約済み	TMR2_postscaled	TMR2_postscaled
00000	T2INPPS で選択されたピン	T4INPPS で選択されたピン	T6INPPS で選択されたピン

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

表 23-5: Timer2 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
CCP1CON	EN	OE	OUT	FMT	MODE<3:0>				259
CCP2CON ⁽²⁾	EN	OE	OUT	FMT	MODE<3:0>				259
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
T2PR	Timer2 モジュール周期レジスタ								232*
TMR2	8 ビット TMR2 レジスタ用の保持レジスタ								232*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				247
T2CLKCON	—	—	—	—	CS<3:0>				246
T2RST	—	—	—	—	RSEL<3:0>				249
T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				248	
T4PR	Timer4 モジュール周期レジスタ								232*
TMR4	8 ビット TMR4 レジスタ用の保持レジスタ								232*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				247
T4CLKCON	—	—	—	—	CS<3:0>				246
T4RST	—	—	—	—	RSEL<3:0>				249
T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				248	
T6PR	Timer6 モジュール周期レジスタ								232*
TMR6	8 ビット TMR6 レジスタ用の保持レジスタ								232*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				247
T6CLKCON	—	—	—	—	CS<3:0>				246
T6RST	—	—	—	—	RSEL<3:0>				249
T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				248	

凡例: — = 未実装、「0」として読み出し。網掛けの部分は Timer2 モジュールでは使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9 のみです。

24.0 キャプチャ/コンペア/PWM モジュール

キャプチャ/コンペア/PWM モジュールは、各種イベントのタイミング計測/制御とパルス幅変調 (PWM) 信号生成のための周辺モジュールです。キャプチャモードでは、イベント継続時間を計測できます。コンペアモードでは、あらかじめ設定した時間が経過した時点で外部イベントをトリガできます。PWM モードでは、各種周波数とデューティ サイクルのパルス幅変調信号を生成できます。

表 24-1: 利用可能な CCP モジュール

デバイス	CCP1	CCP2
PIC16(L)F1764	•	•
PIC16(L)F1765	•	•
PIC16(L)F1768	•	•
PIC16(L)F1769	•	•

Note 1: CCP モジュールを複数内蔵したデバイスでは、レジスタ名に十分注意を払う必要があります。各 CCP モジュールはモジュール略称の末尾にある数字で区別します。例えば、CCP1CON と CCP2CON はレジスタの機能自体は同じで、それぞれ CCP1 と CCP2 の動作を制御します。

2: このセクションで CCP と総称するモジュールの動作は、いずれも CCPx モジュールの全てに等しく適用されます。レジスタ名、モジュール信号、I/O ピン、ビット名の総称には「x」を使います。必要に応じて、各モジュールに該当する数字を入れます。

24.1 キャプチャモード

このセクションで説明するキャプチャモード機能は全ての CCPx モジュールで利用でき、その機能は同じです。

キャプチャモードでは 16 ビットの Timer1 リソースを使います。CCPx 入力でイベントが発生すると TMR1H:TMR1L レジスタペアの 16 ビット値がキャプチャされ、16 ビットの CCPRxH:CCPRxL レジスタペアに格納されます。イベントの定義は以下のいずれかであり、CCPxCON レジスタの MODE<3:0> ビットで設定します。

- 全てのエッジ (立ち上がりりと立ち下がり)
- 全ての立ち下がりエッジ
- 全ての立ち上がりエッジ
- 立ち上がりエッジ 4 回ごと
- 立ち上がりエッジ 16 回ごと

CCPx キャプチャ入力信号は、CCPxCAP レジスタの CTS ビットで以下の選択肢を使って設定します。

- CCPx ピン
- コンパレータ 1 出力 (C1_OUT_sync)
- コンパレータ 2 出力 (C2_OUT_sync)
- コンパレータ 3 出力 (C3_OUT_sync)
- コンパレータ 4 出力 (C4_OUT_sync)
- LC2_output
- LC3_output
- 状態変化割り込みトリガ (IOC_interrupt)

キャプチャが実行されると、PIRx レジスタの割り込み要求フラグビット CCPxIF がセットされます。この割り込みフラグはソフトウェアでクリアする必要があります。CCPRxH:CCPRxL レジスタペアの値を読み出す前に再度キャプチャが実行されると、以前にキャプチャした値が新しいキャプチャ値で上書きされます。

図24-1に、キャプチャ動作の概略ブロック図を示します。

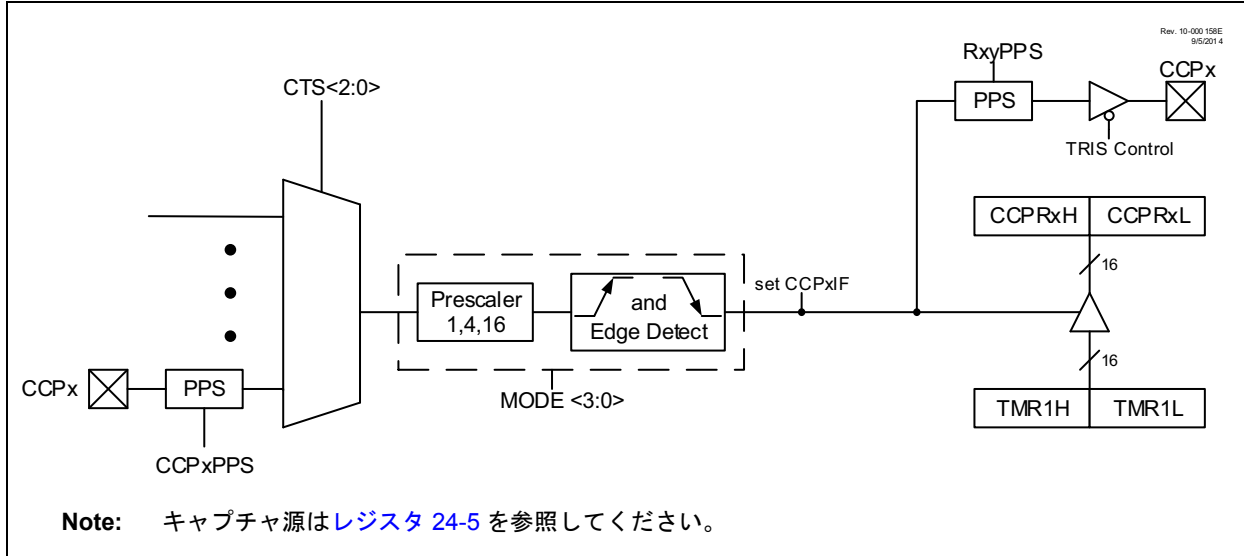
24.1.1 CCP ピンの設定

キャプチャモードでは、CCPxCAP レジスタの CTS ビットを使って割り込み要因を選択します。CCPx ピンが選択されている場合、対応する TRIS 制御ビットをセットしてこのピンを入力に設定する必要があります。

Note: CCPx ピンを出力に設定した場合、そのポートへの書き込み動作を実行するとキャプチャ条件が発生します。

PIC16(L)F1764/5/8/9

図 24-1: キャプチャモード動作のブロック図



24.1.2 Timer1 モードのリソース

CCP モジュールのキャプチャ機能を使うには、Timer1 の動作をタイマモードまたは同期カウンタモードに設定する必要があります。非同期カウンタモードの場合、キャプチャ機能は使えません。

Timer1 の設定の詳細は[セクション 22.0「Timer1/3/5 モジュール \(ゲート制御対応\)」](#)を参照してください。

24.1.3 ソフトウェア割り込みモード

キャプチャモードを変更すると、誤ったキャプチャ割り込みが発生する可能性があります。このような割り込みを防ぐため、PIEx レジスタの CCPxIE 割り込みイネーブルビットをクリアしておく必要があります。また、動作モード変更後には PIRx レジスタの CCPxIF 割り込みフラグビットをクリアする必要があります。

Note: キャプチャモードでは、Timer1 のクロック源をシステムクロック (Fosc) としてはいけません。キャプチャモードで CCPx ピンのトリガイベントを認識するには、Timer1 のクロック源を命令クロック (Fosc/4) または外部クロック源とする必要があります。

24.1.4 CCP プリスケーラ

プリスケーラの設定は 4 種類あり、CCPxCON レジスタの MODE<3:0> ビットで設定します。CCP モジュールが OFF の場合、またはキャプチャモード以外の場合、プリスケーラ カウンタはクリアされます。何らかのリセットが発生するとプリスケーラ カウンタはクリアされます。

キャプチャ プリスケーラを切り換えてもプリスケーラはクリアされないため、不正な割り込みが生成される事があります。このような予期しない割り込みを防ぐため、プリスケーラを変更する前に CCPxCON レジスタの EN ビットをクリアしてモジュールを OFF にします。

24.1.5 スリープ中のキャプチャ

キャプチャモードの動作は、Timer1 モジュールにより決まります。キャプチャモードでは、Timer1 モジュールの駆動方法として 2 通りの設定が可能です。命令クロック (Fosc/4) または外部クロック源を使って駆動できます。

Fosc/4 をクロック源に使うと、Timer1 はスリープ中にインクリメントしません。デバイスがスリープから復帰すると、Timer1 はスリープ直前の状態から動作を再開します。

外部クロック源で Timer1 が動作している場合、スリープ中もキャプチャモードは動作を継続します。

24.1.6 代替ピンの配置

このモジュールで使う I/O ピンは、PPS 制御を使って別のピンへ割り当てられます。詳細は[セクション 12.0「ペリフェラル ピンセレクト \(PPS\) モジュール」](#)を参照してください。

24.1.7 キャプチャ出力

キャプチャが発生すると常にシステムクロック 1 周期分 (1/Fosc) の間 CCP の出力が High になります。この出力は以下の周辺モジュールの入力信号として利用できます。

- ADC のトリガ
- COG
- PRG
- DSM
- CLC
- オペアンプのオーバーライド
- Timer2/4/6 のリセット
- 任意のデバイスピン

さらに、CCP 出力は PPS 制御で任意のピンから出力できます。

24.2 コンペアモード

このセクションで説明するコンペアモード機能は、全ての CCP モジュールで利用でき、その機能は同じです。コンペアモードでは 16 ビットの Timer1 リソースを使います。CCPRxH:CCPRxL レジスタペアの 16 ビット値と TMR1H:TMR1L レジスタペアの 16 ビット値を常に比較します。両方の値が一致した時、以下のイベントのいずれかを発生させる事ができます。

- CCPx 出力をトグルする
- CCPx 出力をセットする
- CCPx 出力をクリアする
- CCPx 出力にパルスを生成する
- ソフトウェア割り込みを生成する
- 自動変換トリガ

ピンの動作は CCPxCON レジスタの MODE<3:0> 制御ビットの値で決まります。同時に、割り込みフラグ CCPxIF ビットがセットされます。

全てのコンペアモードで割り込みを生成できます。

図 24-2 に、コンペア動作の概略ブロック図を示します。

24.2.1 自動変換トリガ

自動変換トリガモード (CCPxM<3:0> = 1011) を選択した場合、CCPx モジュールは以下のように動作します。

- Timer1 をリセットする
- ADC が有効な場合、A/D 変換を開始する

このモードの CCPx モジュールは CCPx ピンの出力を制御しません。

CCP の自動変換トリガは、TMR1H:TMR1L レジスタペアと CCPRxH:CCPRxL レジスタペアが一致するとすぐに出力されます。TMR1H:TMR1L レジスタペアは、

Timer1 クロックの次の立ち上がりエッジまでリセットされません。ADC モジュールが有効な場合、自動変換トリガ出力によって A/D 変換が開始します。これにより、CCPRxH:CCPRxL レジスタペアを Timer1 の 16 ビット プログラマブル周期レジスタとして使います。詳細は [セクション 16.2.5 「自動変換トリガ」](#) を参照してください。

Note 1: CCP モジュールからの自動変換トリガは、PIR1 レジスタの割り込みフラグビット TMR1IF をセットしません。

2: 自動変換トリガが発生したクロックエッジの後、Timer1 がリセットされるクロックエッジまでの間に CCPRxH:CCPRxL レジスタペアの内容を変更して一致状態を解消すると、リセットは発生しません。

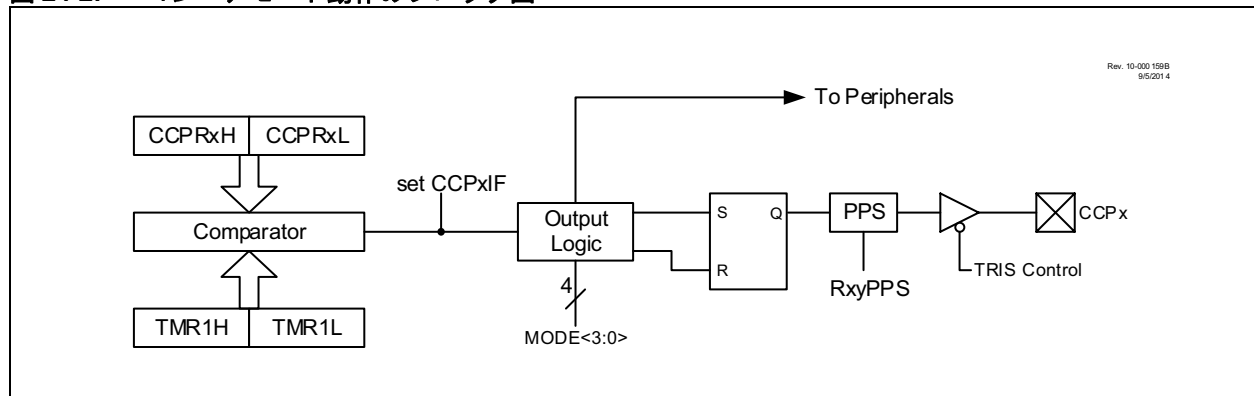
24.2.2 CCPx ピンの設定

コンペアモードでは、対応する TRIS ビットをクリアして CCPx ピンを出力に設定する必要があります。

また、PPS 制御を使って CCPx ピンの機能を他のピンに移動できます。詳細は [セクション 12.0 「ペリフェラルピンセレクト \(PPS\) モジュール」](#) を参照してください。

Note: CCPxCON レジスタをクリアすると、CCPx コンペア出力ラッチは既定値の Low レベルに変更されます。これは PORT I/O データラッチではありません。

図 24-2: コンペアモード動作のブロック図



PIC16(L)F1764/5/8/9

24.2.3 Timer1 モードのリソース

コンペアモードでは、Timer1 がタイマモードまたは同期カウンタモードのどちらかで動作している必要があります。非同期カウンタモードの場合、コンペア機能は使えません。

Timer1 の設定の詳細は[セクション 22.0「Timer1/3/5 モジュール \(ゲート制御対応\)」](#)を参照してください。

Note: コンペアモードでは、Timer1 のクロック源としてシステムクロック (Fosc) を使ってはいけません。コンペアモードで CCPx ピンのトリガイベントを認識するには、Timer1 のクロック源を命令クロック (Fosc/4) または外部クロック源とする必要があります。

24.2.4 ソフトウェア割り込みモード

ソフトウェア割り込み生成モード (MODE<3:0> = 1010) を選択した場合、CCPx モジュールは CCPx ピンの出力を制御しません (CCPxCON レジスタ参照)。

24.2.5 スリープ中のコンペア

コンペアモードは、システムクロック (Fosc) を使って動作します。Fosc はスリープ中停止するため、コンペアモードはスリープ中には機能しません。

24.2.6 代替ピンの配置

このモジュールで使う I/O ピンは、PPS 制御を使って別のピンへ割り当てられます。詳細は[セクション 12.0「ペリフェラル ピンセレクト \(PPS\) モジュール」](#)を参照してください。

24.2.7 キャプチャ出力

コンペアモードでは、CCPRxH:CCPRxL レジスタペアの 16 ビット値が TMR1H:TMR1L レジスタペアと一致すると CCP が出力を生成します。コンペア出力は CCP のコンペアモード設定によって異なります。CCPxCON レジスタの MODE ビットが「1011」または「1010」と等しい場合、TMR1 が CCPRxH:CCPRxL レジスタペアと等しい間 CCP は High を出力します。これはパルス幅が TMR1 プリスケラで決まる事を意味します。CCPxCON の MODE ビットが「0001」または「0010」と等しい場合、一致の時に出力が「0」から「1」(またはその逆)にトグルされます。CCPxCON の MODE ビットが「1001」と等しい場合、一致の時に出力はクリアされ、MODE ビットが「1000」と等しい場合、一致の時に出力がセットされます。この出力信号は、以下の周辺モジュールで使う事ができます。

- ADC のトリガ
- COG
- PRG
- DSM
- CLC
- オペアンプのオーバーライド
- Timer2/4/6 のリセット
- 任意のデバイスピン

24.3 PWM の概要

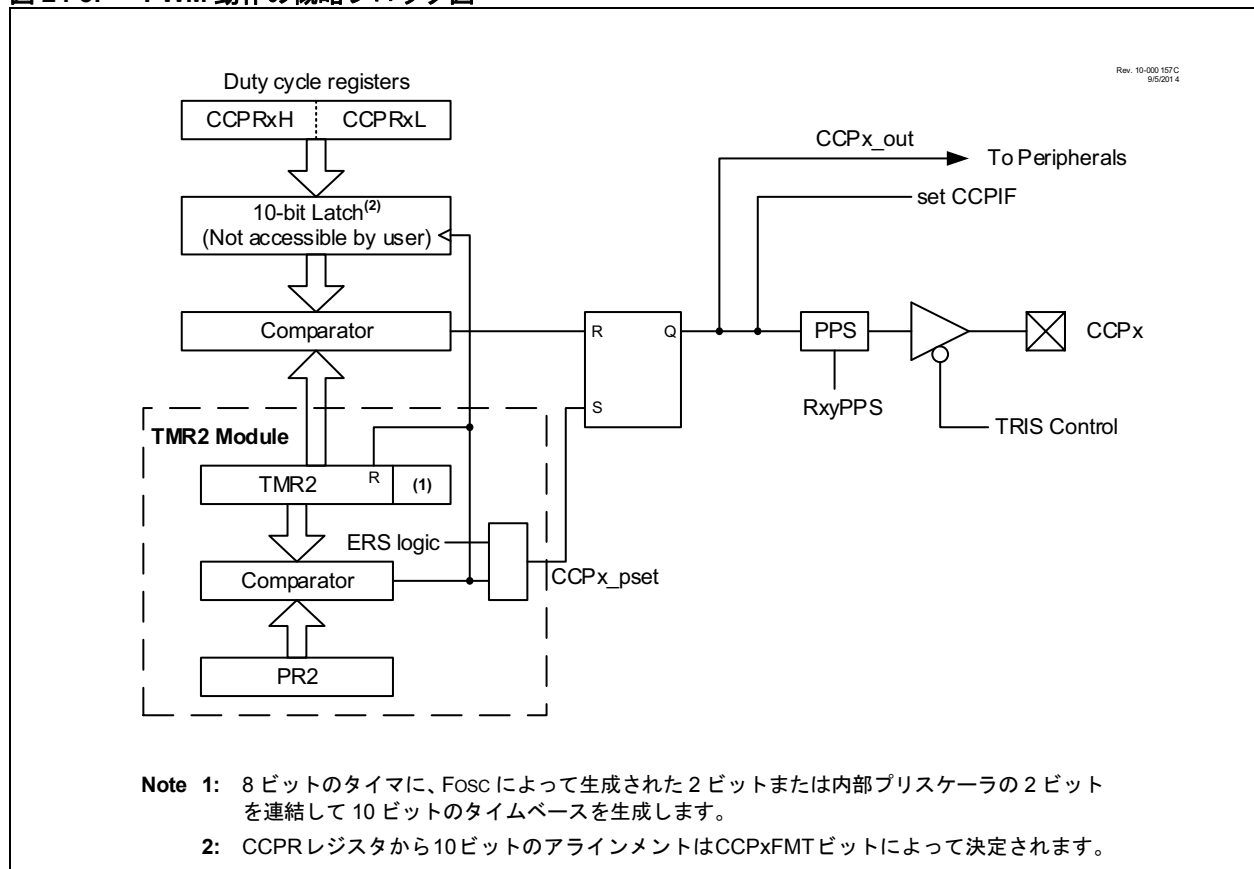
PWM(パルス幅変調)は完全 ON と完全 OFF の状態を切り換えながら負荷に電力を供給する仕組みです。PWM 信号は矩形波に似ており、信号の High の部分が ON、Low の部分が OFF です。High の部分はパルス幅とも呼ばれ、その時間の長さをステップ単位で変化させる事ができます。ステップ数を多くするとパルス幅が広くなり、負荷に供給される電力が増加します。ステップ数を少なくするとパルス幅が狭くなり、負荷に供給される電力が減少します。1 サイクル全体 (すなわち ON 時間と OFF 時間の合計) を PWM 周期といいます。

1 回の PWM 周期に設定できる最大ステップ数を PWM 分解能といいます。PWM 分解能が高いほどパルス幅の時間、すなわち負荷に供給される電力を細かく制御できます。

PWM 周期全体に対する ON 時間の割合を % で表したものをデューティ サイクルと呼び、0% なら完全 OFF、100% なら完全 ON を表します。デューティ サイクルの値が小さいほど負荷に供給される電力が少なく、値が大きいくほど多くなります。

図 24-3 に、PWM 信号の代表的な波形を示します。

図 24-3: PWM 動作の概略ブロック図



PIC16(L)F1764/5/8/9

24.3.1 標準型 PWM の動作

このセクションで説明する標準型 PWM 機能は、全ての CCP モジュールで利用でき、その機能は同じです。

標準型 PWM モードでは、最高分解能 10 ビットの PWM 信号を CCPx ピンに生成します。周期、デューティ サイクル、分解能は以下のレジスタで設定します。

- T2PR/T4PR/T6PR レジスタ
- T2CON/T4CON/T6CON レジスタ
- CCPRxH:CCPRxL レジスタペア

図 24-3 に、PWM 動作の概略ブロック図を示します。

Note 1: CCPx ピンへの PWM 出力を有効にするには、対応する TRIS ビットをクリアする必要があります。

2: CCPxCON レジスタをクリアすると、CCP モジュールは CCPx ピンを制御しません。

24.3.2 PWM 動作の設定

CCP モジュールを標準型 PWM 動作に設定するには、以下のステップを実行します。

1. 対応する TRIS ビットをセットして CCPx ピン出力ドライバを無効にする。
2. CCPTMRS レジスタを設定して PWM に対応するタイマを選択する。
3. 対応する T2PR/T4PR/T6PR レジスタに PWM 周期の値を書き込む。
4. CCPxCON レジスタに適切な値を書き込んで CCP モジュールを PWM モードに設定する。
5. PWM デューティ サイクルの値を CCPRxH:CCPRxL レジスタペアに書き込む。
6. ステップ 2 で選択したタイマを設定してスタートさせる。
 - PIRx レジスタのタイマ割り込みフラグビットをクリアする (下記 Note 参照)。
 - TxCON レジスタの CKPS ビットにタイマのプリスケール値を設定する。
 - TxCON レジスタの ON ビットをセットしてタイマを有効にする。
7. PWM 出力ピンを有効にする。
 - タイマがオーバーフローして PIRx レジスタのタイマ割り込みビットがセットされるのを待つ (下記 Note 参照)。
 - 対応する TRIS ビットをクリアして CCPx ピン出力ドライバを有効にする。

Note: 最初の PWM 出力から完全なデューティ サイクルと周期で出力するには、セットアップ シーケンスに上記のステップを含める必要があります。完全な PWM 信号で開始する必要がない場合、ステップ 6 を省略できます。

24.4 CCP/PWM クロックの選択

PIC16(L)F1764/5/8/9 では、各 CCP および PWM モジュールをどのタイマで制御するかを個別に選択できます。モジュールごとに個別の設定が可能です。

最大3個の自動再書き込み付き8ビットタイマ(Timer2/4/6)が利用できます。CCPおよびPWMモジュールのPWMモードは、これらのタイマをどれでも使う事ができます。

どのタイマを使うかは CCPTMRS レジスタで選択します。

24.4.1 CCP モジュールでの TMR2/4/6 の使用

本デバイスは TMR2 モジュールの新規バージョンを内蔵して新しいモードを多く備えているため、従来品より多様な PWM 信号のカスタマイズと制御が可能です。Timer2 の各種モードを使った PWM 信号生成の例は [セクション 23.5「動作例」](#) を参照してください。CCP 動作では、PWM タイムベースとして使うタイマで Fosc/4 クロック源を選択する必要があります。

24.4.2 PWM の周期

PWM の周期は Timer2/4/6 の T2PR/T4PR/T6PR レジスタで指定します。PWM の周期は [式 24-1](#) で求める事ができます。

式 24-1: PWM の周期

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (TMR2\ Prescale\ Value)$$

Note 1: $T_{osc} = 1/F_{osc}$

TMR2/4/6 と対応する T2PR/T4PR/T6PR レジスタがそれぞれ等しくなると、直後のインクリメント サイクルで以下の3つのイベントが実行されます。

- TMR2/4/6 がクリアされる
- CCPx ピンがセットされる (例外: PWM のデューティ サイクル = 0% の場合、ピンはセットされない)
- PWM のデューティ サイクルは CCPRxH:CCPRxL ペアから内蔵 10 ビットラッチにラッチされる

Note: PWM の周波数は、タイマのポストスケアラ ([図 24-1](#) 参照) に関係なく決定します。

24.4.3 PWM のデューティ サイクル

PWM デューティ サイクルは2個のレジスタ (CCPRxH:CCPRxL レジスタペア) に 10 ビット値を書き込む事により指定します。個々のビットの書き込み先は CCPxCON レジスタの FMT ビットで決定します。FMT = 0 の場合、デューティ サイクル値の上位 2 ビットは CCPRxH レジスタのビット <1:0> に、残り 8 ビットは CCPRxL レジスタに書き込まれます。FMT = 1 の場合、デューティ サイクル値の下位 2 ビットは CCPRxL レジスタのビット <7:6> に、上位 8 ビットは CCPRxH レジスタに書き込まれます。これを図解したものが [図 24-4](#) です。これらのビットはいつでも書き込む事ができます。デューティ サイクルの値は、PWM 周期が完了

する (すなわち T2PR/T4PR/T6PR レジスタと TMR2/4/6 レジスタが一致する時点) まで内蔵ラッチにラッチされません。

PWM のパルス幅は [式 24-2](#) で求めます。PWM のデューティ サイクル比は [式 24-3](#) で求めます。

式 24-2: パルス幅

$$Pulse\ Width = CCPRxH:CCPRxL \cdot T_{osc} \cdot (TMR2\ Prescale\ Value)$$

式 24-3: デューティ サイクル比

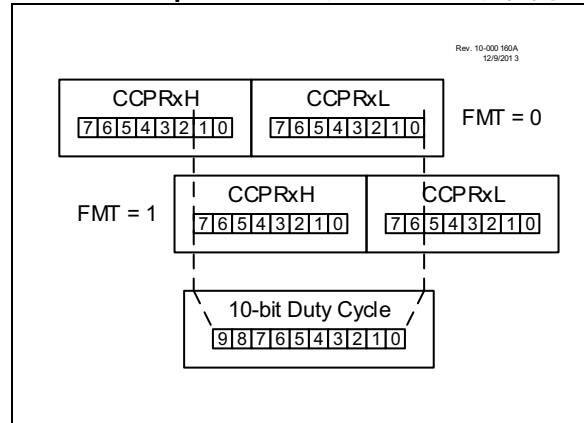
$$Duty\ Cycle\ Ratio = \frac{CCPRxH:CCPRxL}{4(PR_x + 1)}$$

PWM デューティ サイクルレジスタは、PWM 動作でグリッチが発生しないようにダブルバッファ構成になっています。

8 ビットのタイマ TMR2/4/6 レジスタに、2 ビットの内部システムクロック (Fosc) または 2 ビットのプリスケアラを連結して、10 ビットのタイムベースを生成します。Timer2/4/6 のプリスケアラが 1:1 に設定されている場合はシステムクロックが使用されます。

10 ビットのタイムベースが内蔵バッファレジスタと一致すると、CCPx ピンがクリアされます ([図 24-3](#) 参照)。

図 24-4: ccpx デューティ サイクルの位置調整



PIC16(L)F1764/5/8/9

24.4.4 PWM の分解能

分解能は、1 周期に設定可能なデューティ サイクルの数を決定します。例えば分解能 10 ビットの場合は 1024 通りのデューティ サイクル、分解能 8 ビットの場合は 256 通りのデューティ サイクルを設定できます。

PWM 分解能の最大値は 10 ビットです (T2PR/T4PR/T6PR が 255 の時)。式 24-4 に示すように、分解能は T2PR/T4PR/T6PR レジスタの値の関数です。

式 24-4: PWM 分解能

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)} \text{ bits}$$

Note: パルス幅の値が周期より大きい場合、対応する PWM ピンの状態は変化しません。

表 24-2: PWM の周波数と分解能の例 (Fosc = 20 MHz)

PWM の周波数	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
タイマのプリスケール値	16	4	1	1	1	1
T2PR の値	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最高分解能 (ビット)	10	10	10	8	7	6

表 24-3: PWM の周波数と分解能の例 (Fosc = 8 MHz)

PWM の周波数	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
タイマのプリスケール値	16	4	1	1	1	1
T2PR の値	0x65	0x65	0x65	0x19	0x0C	0x09
最高分解能 (ビット)	8	8	8	6	5	5

24.4.5 システムクロック周波数の変更

PWM の周波数はシステムクロック周波数から生成されます。システムクロック周波数を変更すると、PWM の周波数が変化します。詳細は、[セクション 5.0「オシレータ モジュール \(フェイルセーフクロックモニタ機能付き\)」](#) を参照してください。

24.4.6 リセットの影響

何らかのリセットが発生すると全てのポートが入力モードとなり、CCP レジスタはリセット状態に戻ります。

24.4.7 PWM 出力

PWM モードにおける CCP の出力は、モジュールで生成された上記の PWM 信号です。この出力信号は、以下の周辺モジュールで使う事ができます。

- ADC のトリガ
- COG
- PRG
- DSM
- CLC
- オペアンプのオーバーライド
- Timer2/4/6 のリセット
- 任意のデバイスピン

24.5 レジスタ定義 : CCP 制御

レジスタ 24-1: CCPxCON: CCPx 制御レジスタ

R/W-0/0	U-0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	
EN	—	OUT	FMT	MODE<3:0>				
bit 7								bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7 **EN:** CCPx モジュール イネーブルビット

1 = CCPx を有効にする
0 = CCPx を無効にする

bit 6 **未実装:** 「0」として読み出し

bit 5 **OUT:** CCPx 出力データビット (読み出し専用)

bit 4 **FMT:** CCPW(パルス幅)アラインメント ビット

MODE = PWM モードの場合

1 = 左詰めフォーマット (CCPRxH<7> が PWM デューティ サイクルの MSb)
0 = 右詰めフォーマット (CCPRxL<0> が PWM デューティ サイクルの LSb)

bit 3-0 **MODE<3:0>:** CCPx モード選択ビット

11xx = PWM モード

1011 = コンペアモード: パルス出力、TMR1 をクリア

1010 = コンペアモード: パルス出力 (0 - 1 - 0)

1001 = コンペアモード: コンペア一致で出力をクリアする。出力はこのモードを選択した時にセットされる

1000 = コンペアモード: コンペア一致で出力をセットする。出力はこのモードを選択した時にクリアされる

0111 = キャプチャモード: 16 回目の立ち上がりエッジごと

0110 = キャプチャモード: 4 回目の立ち上がりエッジごと

0101 = キャプチャモード: 全ての立ち上がりエッジ

0100 = キャプチャモード: 全ての立ち下がりエッジ

0011 = キャプチャモード: 全ての立ち上がり立ち下がりエッジ

0010 = コンペアモード: 一致時に出力をトグルする

0001 = コンペアモード: 一致時に出力をトグルし、TMR1 をクリアする

0000 = キャプチャ/コンペア/PWM OFF (CCPx モジュールをリセット) (下位互換のため予約済み)

PIC16(L)F1764/5/8/9

レジスタ 24-2: CCPTMRS: PWM タイマ選択制御 0 レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
P4TSEL<1:0> ⁽¹⁾		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-6 **P4TSEL<1:0>**: PWM4 タイマ選択ビット ⁽¹⁾
11 = 予約済み
10 = PWM4 の PWM モードのタイマとして Timer6 を使う
01 = PWM4 の PWM モードのタイマとして Timer4 を使う
00 = PWM4 の PWM モードのタイマとして Timer2 を使う
- bit 5-4 **P3TSEL<1:0>**: PWM3 タイマ選択ビット
11 = 予約済み
10 = PWM3 の PWM モードのタイマとして Timer6 を使う
01 = PWM3 の PWM モードのタイマとして Timer4 を使う
00 = PWM3 の PWM モードのタイマとして Timer2 を使う
- bit 3-2 **C2TSEL<1:0>**: CCP2 (PWM2) タイマ選択ビット
11 = 予約済み
10 = CCP2 の PWM モードのタイマとして Timer6 を使う
01 = CCP2 の PWM モードのタイマとして Timer4 を使う
00 = CCP2 の PWM モードのタイマとして Timer2 を使う
- bit 1-0 **C1TSEL<1:0>**: CCP1 (PWM1) タイマ選択ビット
11 = 予約済み
10 = CCP1 の PWM モードのタイマとして Timer6 を使う
01 = CCP1 の PWM モードのタイマとして Timer4 を使う
00 = CCP1 の PWM モードのタイマとして Timer2 を使う

レジスタ 24-3: CCPRxL: CCPx 下位バイトレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPR<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 MODE = キャプチャモード
CCPRxL<7:0>: キャプチャされた TMR1 値の LSB
MODE = コンペアモード
CCPRxL<7:0>: TMR1 値とコンペアされた LSB
MODE = PWM モード && FMT = 0
CCPRxL<7:0>: CCPW<7:0> — パルス幅の下位 8 ビット
MODE = PWM モード && FMT = 1
CCPRxL<7:6>: CCPW<1:0> — パルス幅の下位 2 ビット
CCPRxL<5:0>: 未使用

レジスタ 24-4: CCPRxH: CCPx 上位バイトレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPR<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 MODE = キャプチャモード
CCPRxH<7:0>: キャプチャされた TMR1 値の MSB
MODE = コンペアモード
CCPRxH<7:0>: TMR1 値とコンペアされた MSB
MODE = PWM モード && FMT = 0
CCPRxH<7:2>: 未使用
CCPRxH<1:0>: CCPW<9:8> — パルス幅の上位 2 ビット
MODE = PWM モード && FMT = 1
CCPRxH<7:0>: CCPW<9:2> — パルス幅の上位 8 ビット

PIC16(L)F1764/5/8/9

レジスタ 24-5: CCPxCAP: CCPx キャプチャ入力選択レジスタ

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-3 **未実装:** 「0」として読み出し
bit 2-0 **CTS<2:0>:** キャプチャトリガ入力選択ビット
111 = IOC_event
110 = LC3_output
101 = LC2_output
100 = C4_sync_out⁽¹⁾
011 = C3_sync_out⁽¹⁾
010 = C2_sync_out
001 = C1_sync_out
000 = CCPxPPS レジスタで選択したピン

Note 1: PIC16(L)F1768/9 のみです。PIC16(L)F1764/5 では未実装です。

表 24-4: 標準型 PWM 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
CCPxCAP	—	—	—	—	—	CTS<2:0>			262
CCPxCON	EN	OE	OUT	FMT	MODE<3:0>				259
CCPRxL	キャプチャ/コンペア/PWM レジスタ x (LSB)								261
CCPRxH	キャプチャ/コンペア/PWM レジスタ x (MSB)								261
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		260
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
T2PR	Timer2 周期レジスタ								
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				250
TMR2	Timer2 モジュール レジスタ								250
T4PR	Timer4 周期レジスタ								250*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				250
TMR4	Timer4 モジュール レジスタ								250
T6PR	Timer6 周期レジスタ								250*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				250
TMR6	Timer6 モジュール レジスタ								250

凡例: — = 未実装、「0」として読み出し。網掛けの部分は PWM では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

24.6 CCP/PWM クロックの選択

本デバイスでは各 CCP および PWM モジュールをどのタイマで制御するかを個別に選択できます。モジュールごとの設定が可能です。

最大3個の自動再書き込み付き8ビットタイマ(Timer2、Timer4、Timer6)が利用できます。CCP および 10 ビット PWM モジュールの PWM モードでは、これらのうち任意のタイマを使う事ができます。

どのタイマを使うかは、CCPTMRS レジスタで選択します。

24.7 レジスタ定義 : CCP/PWM タイマ制御

レジスタ 24-6: CCPTMRS: PWM タイマ選択制御 0 レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7-6 **P4TSEL<1:0>**: PWM4 タイマ選択ビット
11 = 予約済み
10 = PWM4 のタイマとして Timer6 を使う
01 = PWM4 のタイマとして Timer4 を使う
00 = PWM4 のタイマとして Timer2 を使う
- bit 5-4 **P3TSEL<1:0>**: PWM3 タイマ選択ビット
11 = 予約済み
10 = PWM3 のタイマとして Timer6 を使う
01 = PWM3 のタイマとして Timer4 を使う
00 = PWM3 のタイマとして Timer2 を使う
- bit 3-2 **C2TSEL<1:0>**: CCP2 (PWM2) タイマ選択ビット
11 = 予約済み
10 = CCP2 の PWM モードのタイマとして Timer6 を使う
01 = CCP2 の PWM モードのタイマとして Timer4 を使う
00 = CCP2 の PWM モードのタイマとして Timer2 を使う
- bit 1-0 **C1TSEL<1:0>**: CCP1 (PWM1) タイマ選択ビット
11 = 予約済み
10 = CCP1 の PWM モードのタイマとして Timer6 を使う
01 = CCP1 の PWM モードのタイマとして Timer4 を使う
00 = CCP1 の PWM モードのタイマとして Timer2 を使う

25.0 10ビットパルス幅変調 (PWM) モジュール

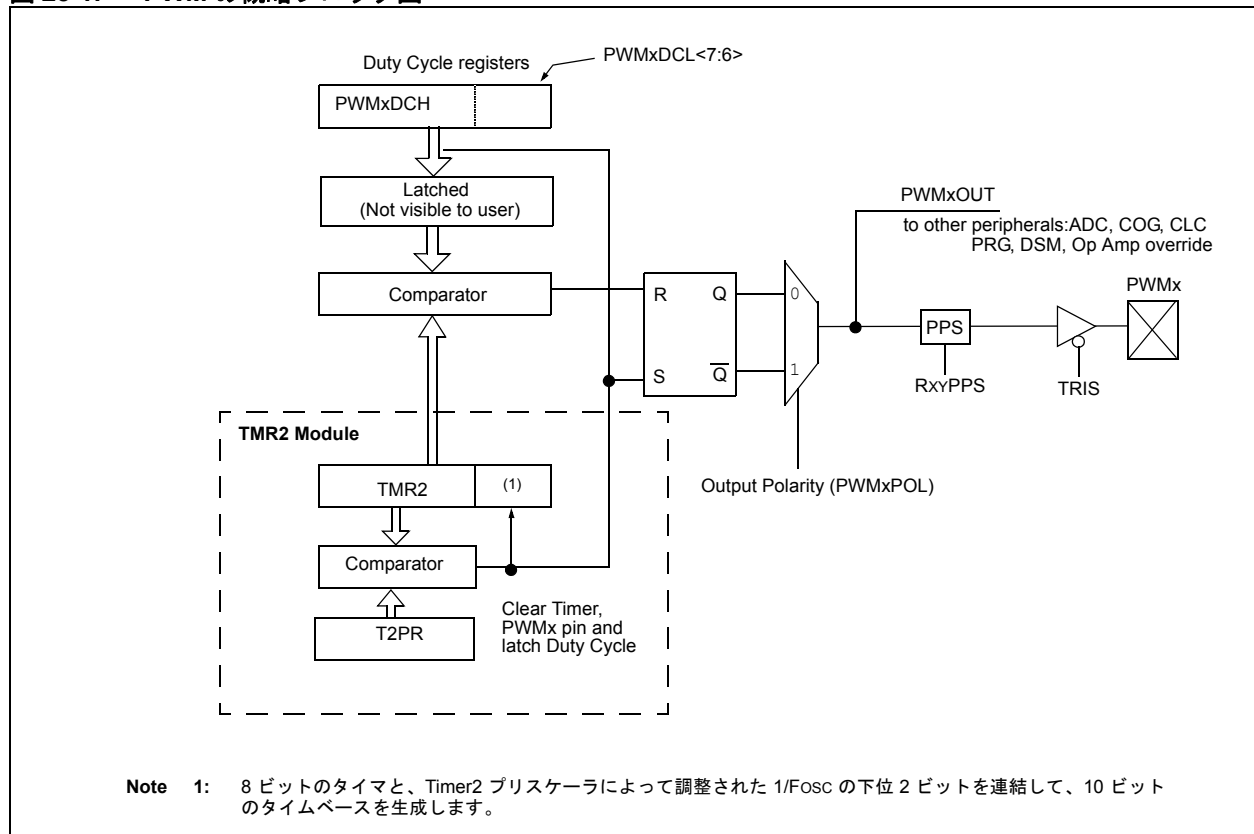
10ビットPWMモジュールは、以下のレジスタで設定するデューティサイクル、周期、分解能によって決まる、パルス幅変調信号を生成します。

- T2PR
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

図 25-1 に、PWM 動作の概略ブロック図を示します。

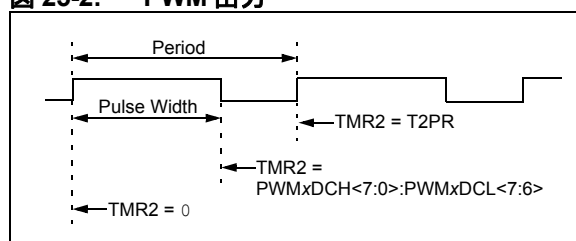
図 25-2 に、PWM 信号の代表的な波形を示します。

図 25-1: PWM の概略ブロック図



このモジュールを PWM 動作用に設定する手順は、[セクション 25.1.9 「PWM 出力を PWMx 出力ピンで使う場合の設定」](#) を参照してください。

図 25-2: PWM 出力



PIC16(L)F1764/5/8/9

25.1 PWMx のピン設定

PWM の出力は全て PORT データラッチと多重化されます。対応する TRIS ビットをクリアしてピンを出力に設定する必要があります。

25.1.1 基本動作

PWM モジュールは分解能 10 ビットの出力を生成します。Timer2 と T2PR で PWM の周期を設定します。PWMxDCL レジスタと PWMxDCH レジスタでデューティサイクルを設定します。周期は全 PWM モジュールで共通ですが、デューティサイクルはモジュールごとに制御されます。

Note: Timer2 のポストスケーラは PWM 周波数の決定には使いません。このポストスケーラは、PWM 出力と異なる周波数のサーボ更新レートを得るために使います。

Timer2 に関連する全 PWM 出力は、TMR2 がクリアされた時点でセットされます。各 PWMx は、対応する PWMxDCH (8MSb) と PWMxDCL<7:6> レジスタ (2LSb) で指定された値と TMR2 が等しくなった時点でクリアされます。この値が T2PR の値以上の場合、PWM 出力は一切クリアされません (100% デューティサイクル)。

Note: DACxREFL レジスタと DACxREFH レジスタはダブルバッファ構成です。これらのバッファは Timer2 と T2PR が一致した時点で更新されます。タイマが一致する前に両方のレジスタが更新されるように注意する必要があります。

25.1.2 PWM 出力の極性

PWMxCON レジスタの PWMxPOL ビットをセットすると、出力の極性が反転します。

25.1.3 PWM 周期

PWM の周期は Timer2 の T2PR レジスタで指定します。PWM の周期は式 25-1 で求める事ができます。

式 25-1: PWM 周期

$$PWM \text{ Period} = [T2PR + 1] \cdot 4 \cdot TOSC \cdot (TMR2 \text{ Prescale Value})$$

Note: TOSC = 1/FOSC

TMR2 と T2PR が等しくなると、直後のインクリメントサイクルで以下の 3 つのイベントが実行されます。

- TMR2 がクリアされる
- PWM 出力がアクティブになる (例外: PWM デューティサイクル=0% の場合、PWM 出力は非アクティブのまま)
- PWMxDCH および PWMxDCL レジスタの値がバッファにラッチされる

Note: Timer2 のポストスケーラは PWM 動作に一切影響を与えません。

25.1.4 PWM デューティ サイクル

PWM のデューティサイクルは、PWMxDCH と PWMxDCL のレジスタペアに 10 ビットの値を書き込んで設定します。PWMxDCH レジスタには上位 8 ビット、PWMxDCL<7:6> には下位 2 ビットを格納します。PWMxDCH レジスタと PWMxDCL レジスタは、いつでも書き込み可能です。

PWM のパルス幅は式 25-2 で求めます。

PWM のデューティサイクル比は式 25-3 で求めます。

式 25-2: パルス幅

$$Pulse \text{ Width} = (PWMxDCH:PWMxDCL<7:6>) \cdot TOSC \cdot (TMR2 \text{ Prescale Value})$$

Note: TOSC = 1/FOSC

式 25-3: デューティ サイクル比

$$Duty \text{ Cycle Ratio} = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(T2PR + 1)}$$

8 ビットタイマの TMR2 レジスタと、Timer2 プリスケーラによって調整された 1/FOSC の下位 2 ビットを連結して、10 ビットのタイムベースを生成します。Timer2 のプリスケーラが 1:1 に設定されている場合、システムクロックが使われます。

25.1.5 PWM 分解能

分解能は、1 周期に設定可能なデューティ サイクルの数を決定します。例えば分解能 10 ビットの場合は 1024 通りのデューティ サイクル、分解能 8 ビットの場合は 256 通りのデューティ サイクルを設定できます。

T2PR が 255 の時、PWM 分解能の最大値は 10 ビットです。式 25-4 に示すように、分解能は T2PR レジスタの値の関数です。

式 25-4: PWM 分解能

Note: パルス幅の値が周期より大きい場合、対応する PWM ピンの状態は変化しません。

表 25-1: PWM の周波数と分解能の例 (Fosc = 20 MHz)

PWM の周波数	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
タイマのプリスケアラ値	64	4	1	1	1	1
T2PR の値	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最高分解能 (ビット)	10	10	10	8	7	6.6

表 25-2: PWM の周波数と分解能の例 (Fosc = 8 MHz)

PWM の周波数	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
タイマのプリスケアラ値	64	4	1	1	1	1
T2PR の値	0x65	0x65	0x65	0x19	0x0C	0x09
最高分解能 (ビット)	8	8	8	6	5	5

25.1.6 スリープ中の動作

スリープ中、TMR2 レジスタはインクリメントせず、モジュールの状態は変化しません。値を駆動している PWMx ピンは、引き続きその値を駆動します。デバイスが復帰すると、TMR2 は直前の状態からインクリメントを再開します。

25.1.7 システムクロック周波数の変更

PWM の周波数はシステムクロック周波数 (Fosc) から生成されます。システムクロック周波数を変更すると、PWM の周波数が変化します。詳細は[セクション 5.0 「オシレータ モジュール \(フェイルセーフクロックモジュール機能付き\)」](#)を参照してください。

25.1.8 リセットの影響

何らかのリセットが発生すると、強制的に全ポートが入力モードとなり、PWM レジスタはリセット状態に戻ります。

25.1.9 PWM 出力を PWMx 出力ピンで使う場合の設定

PWM 出力を PWMx 出力ピンで使う場合、以下の手順で PWM モジュールを設定します。

1. 対応する TRIS ビットをセットして PWMx ピンの出力ドライバを無効にする。
2. PWMxCON レジスタをクリアする。
3. T2PR レジスタに PWM 周期の値を書き込む。
4. PWMxDCH レジスタおよび PWMxDCL レジスタのビット <7:6> に PWM デューティサイクルの値を書き込む。
5. Timer2 を設定して起動する。
 - PIR1 レジスタの TMR2IF 割り込みフラグビットをクリアする (下記 Note 参照)。
 - T2CON レジスタの CKPS ビットに Timer2 のプリスケール値を設定する。
 - T2CON レジスタの ON ビットをセットして Timer2 を有効にする。
6. PWM 出力ピンを有効にして、Timer2 がオーバーフローし、PIR1 レジスタの TMR2IF ビットがセットされるまで待機する (下記 Note 参照)。
7. 対応する TRIS ビットをクリアすると共に、目的のピンの PPS 制御ビットをセットして、PWMx ピンの出力ドライバを有効にする。
8. PWMxCON レジスタに適切な値を書き込んで PWM モジュールを設定する。

Note 1: 最初の PWM 出力から完全なデューティサイクルと周期で出力するには、上記の順番で設定を実行する必要があります。完全な PWM 信号で開始する必要がない場合、ステップ 4 の代わりにステップ 8 を実行します。

2: PWM 出力を他の周辺モジュールでのみ使う場合、PWMx ピン出力を無効にします。

25.1.10 PWM 出力を他の周辺モジュールで使う場合の設定

PWM 出力を他の周辺モジュールで使う場合、以下の手順で PWM モジュールを設定します。

1. 対応する TRIS ビットをセットして PWMx ピンの出力ドライバを無効にする。
2. PWMxCON レジスタをクリアする。
3. T2PR レジスタに PWM 周期の値を書き込む。
4. PWMxDCH レジスタおよび PWMxDCL レジスタのビット <7:6> に PWM デューティサイクルの値を書き込む。
5. Timer2 を設定して起動する。
 - PIR1 レジスタの TMR2IF 割り込みフラグビットをクリアする (下記 Note 参照)。
 - T2CON レジスタの CKPS ビットに Timer2 のプリスケール値を設定する。
 - T2CON レジスタの ON ビットをセットして Timer2 を有効にする。
6. PWM 出力ピンを有効にする。
 - Timer2 がオーバーフローし、PIR1 レジスタの TMR2IF ビットがセットされるまで待機する (下記 Note 参照)。
7. PWMxCON レジスタに適切な値を書き込んで PWM モジュールを設定する。

Note: 最初の PWM 出力から完全なデューティサイクルと周期で出力するには、セットアップシーケンスに上記のステップを含める必要があります。完全な PWM 信号で開始する必要がない場合、ステップ 6 を省略できます。

25.2 レジスタ定義 :10 ビット PWM 制御

表 25-3 に DSM 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1.2.2 「完全ビット名」](#) を参照してください。

表 25-3:

周辺モジュール	ビット名の接頭辞
PWM3	PWM3
PWM4 ⁽¹⁾	PWM4

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 25-1: PWMxCON: PWM 制御レジスタ

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
EN	—	OUT	POL	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7 **EN:** PWM モジュール イネーブルビット
 1 = PWM モジュールを有効にする
 0 = PWM モジュールを無効にする

bit 6 **未実装:** 「0」として読み出し

bit 5 **OUT:** PWM モジュールの出力レベル読み出し用ビット

bit 4 **POL:** PWMx 出力極性選択ビット
 1 = PWM 出力をアクティブ Low にする
 0 = PWM 出力をアクティブ High にする

bit 3-0 **未実装:** 「0」として読み出し

PIC16(L)F1764/5/8/9

レジスタ 25-2: PWMxDCH: PWM デューティ サイクル上位ビット

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
DC<9:2>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
 「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **DC<9:2>**: PWM デューティ サイクル上位ビット
 PWM デューティ サイクルの上位ビットです。下位 2 ビットは PWMxDCL レジスタに格納します。

レジスタ 25-3: PWMxDCL: PWM デューティ サイクル下位ビット

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
DC<1:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
 「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **DC<1:0>**: PWM デューティ サイクル下位ビット
 PWM デューティ サイクルの下位ビットです。上位ビットは PWMxDCH レジスタに格納します。

bit 5-0 **未実装**: 「0」として読み出し

表 25-4: 10 ビット PWM 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0> ⁽¹⁾		C1TSEL<1:0>		264
PWMxCON	EN	—	OUT	POL	MODE<1:0>		—	—	269
PWMxDCH	DC<9:2>								270
PWMxDCL	DC<1:0>		—	—	—	—	—	—	270
RxyPPS	—	—	—	RxyPPS<4:0>					158
TxCON	ON	CKPS<2:0>			OUTPS<3:0>				247
TxCLKCON	—	—	—	—	CS<3:0>				246
TxPR	TMRx 周期レジスタ								232
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例: — = 未実装、「0」として読み出し、u = 不変、x = 未知。網掛けの部分は PWM では使いません。

- Note** 1: 未実装、「1」として読み出します。
 2: PIC16(L)F1768/9 のみです。

26.0 16ビットパルス幅変調 (PWM) モジュール

パルス幅変調 (PWM) モジュールは、以下のレジスタで設定する位相、デューティ サイクル、周期、オフセット イベントカウントによって決まる、パルス幅変調信号を生成します。

- PWMxPH レジスタ
- PWMxDC レジスタ
- PWMxPR レジスタ
- PWMxOF レジスタ

図 26-1 に、PWM 動作の概略ブロック図を示します。

各 PWM モジュールは以下の 4 つの動作モードを持っています。

- 標準
- 一致時にセット
- 一致時にトグル
- センターアライン

各 PWM モードの詳細は[セクション 26.2 「PWM モード」](#)を参照してください。

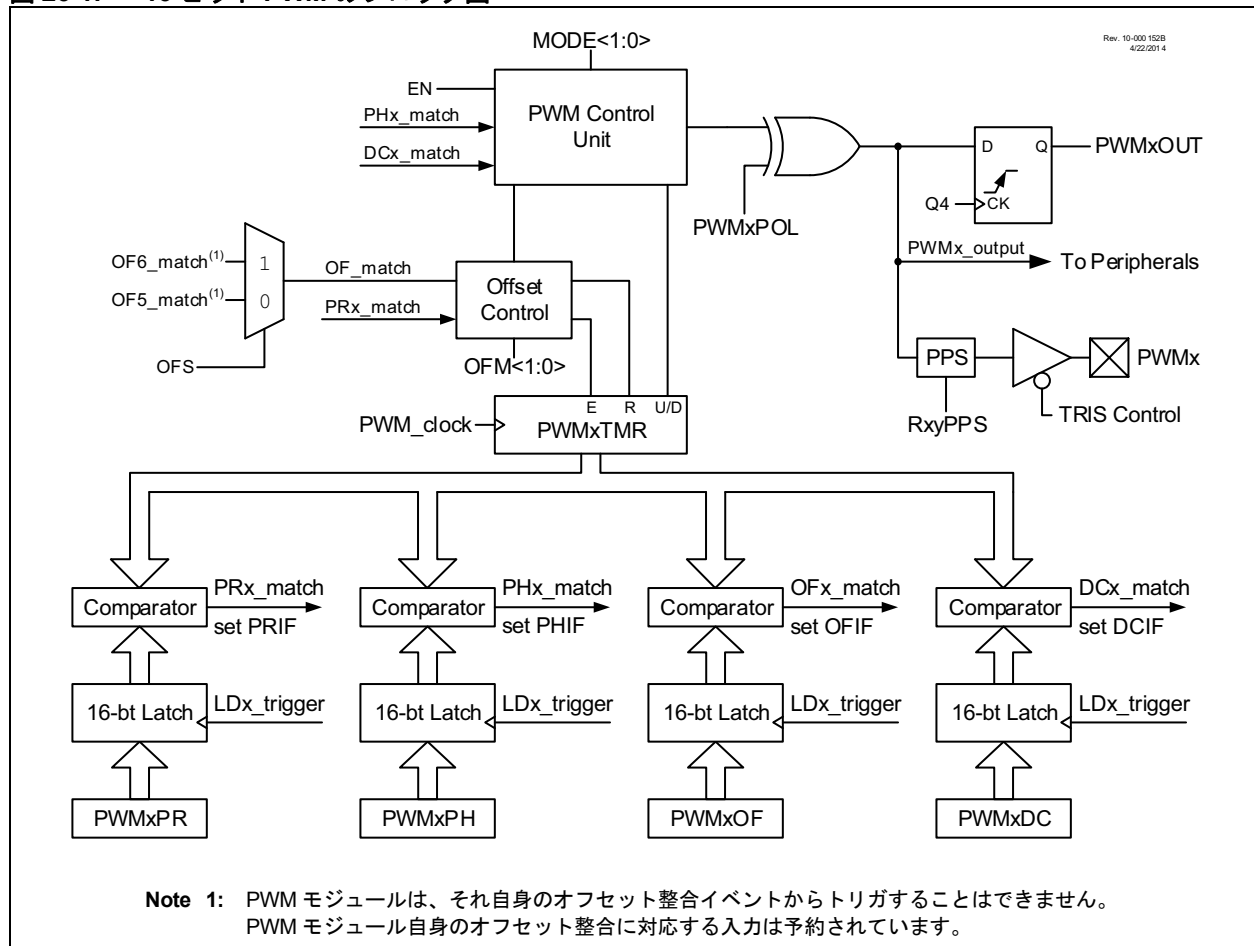
各 PWM モジュールで以下の 4 つのオフセットモードを使えます。

- 独立実行
- 同期スタート付き実行スレーブ
- 同期スタート付きワンショットスレーブ
- 同期スタート、タイマリセット付き連続実行スレーブ

オフセットモードを使うと、他の PWM モジュールに対して波形をずらす事ができます。オフセットモードの詳細は[セクション 26.3 「オフセットモード」](#)を参照してください。

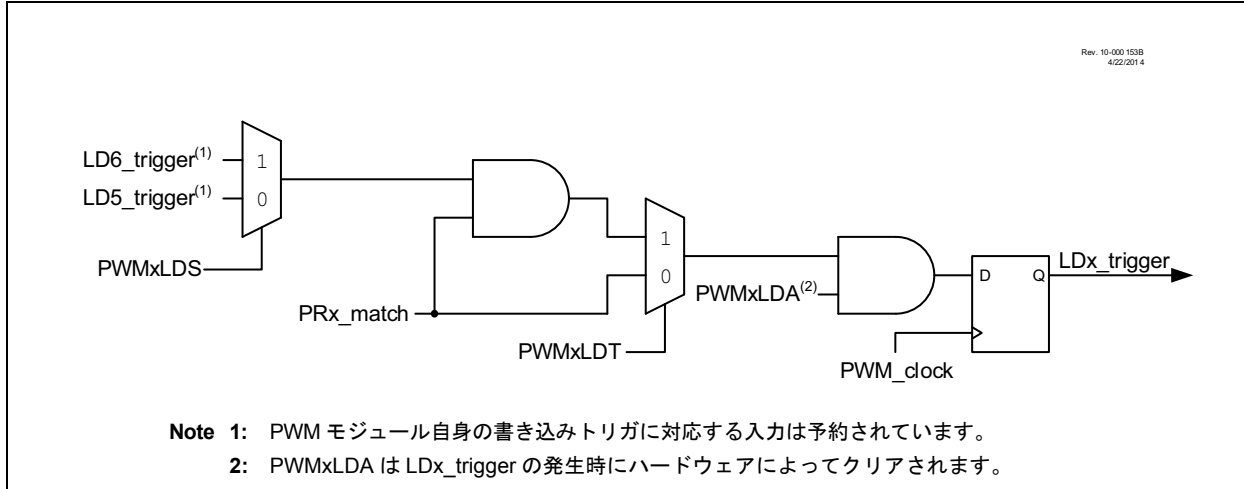
各 PWM モジュールは設定可能な再書き込み動作を備えており、全てのイベントカウントバッファが各周期の末尾で必ず変更されるため、信号のグリッチを回避できます。図 26-2 に、再書き込み動作の概略ブロック図を示します。再書き込み動作の詳細は[セクション 26.4 「再書き込み動作」](#)を参照してください。

図 26-1: 16ビット PWM のブロック図



PIC16(L)F1764/5/8/9

図 26-2: 書き込みトリガのブロック図



26.1 基本動作

PWM モジュールは、分解能 16 ビットのパルス幅変調出力を生成します。

各 PWM モジュールは、PWMxCLKCON レジスタ (レジスタ 26-4) で選択したクロック源で駆動する独立したタイマを備えています。タイマ値はイベントカウンタレジスタと比較され、周期、デューティ サイクル等の PWM 波形の各種イベントを生成します。クロック源のブロック図は図 26-3 を参照してください。

各 PWM モジュールは PWMxCON レジスタの EN ビットで個別に有効化できます。また複数の PWM モジュールを PWMEN レジスタの MPWMxEN ビットで同時に有効化する事もできます。

PWM 出力の状態は PWMxCON レジスタの OUT ビットで読み出す事ができます。モードによってはこのビットはソフトウェアでセットまたはクリアできるため、PWM 波形の高度なソフトウェア制御が可能です。このビットは Fosc/4 に同期しているため、PWM_clock に対してリアルタイムでは変化しません。

Note: PWM_clock > Fosc/4 の場合、OUT ビットは PWM の出力状態を正確には表しません。

26.1.1 PWMx ピンの設定

本デバイスでは周辺モジュールを任意のデバイス I/O ピンに接続するために PPS 機能を使います。RxyPPS 制御レジスタ (レジスタ 12-2) を使って、PWM 出力に対してデバイスのピンを選択します。

全 PWM 出力は PORT データラッチと多重化されているため、対応する PORT TRIS ビットをクリアしてピンを出力として設定する必要があります。

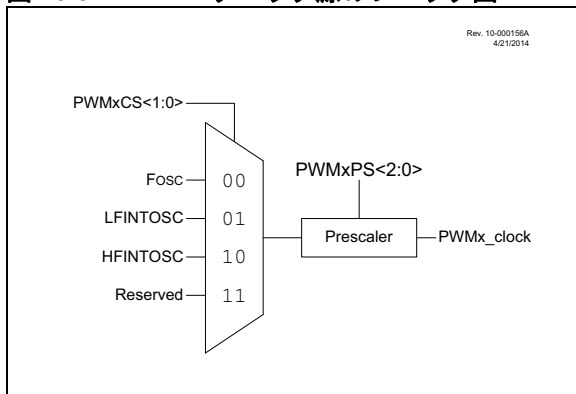
PWM出力で用いるスルーレートを最適化するために、必要に応じてスルーレート機能を設定します。対応する PORT SLRCON ビットをクリアすると高速の出力スイッチングを得る事ができます。

PWM 出力は対応する PORT ODCON ビットをセットするとオーブンドレイン出力に設定できます。

26.1.2 PWMx 出力の極性

PWMxCON レジスタの POL ビットをセットすると、出力の極性が反転します。モジュールが無効の場合でも、極性制御は PWM 出力に影響を与えます。

図 26-3: PWM クロック源のブロック図



26.2 PWM モード

PWM モードは PWMxCON レジスタ (レジスタ 26-1) の MODE<1:0> ビットで選択します。

また全 PWM モードで、3 つのオフセットモードで PWMxTMR を同期させるためにオフセット一致イベントを使う事ができます。詳細は [セクション 26.3「オフセットモード」](#) を参照してください。

26.2.1 標準モード

標準モード (MODE = 00) は単相の PWM 出力を選択します。このモードの PWM 出力は、周期、デューティサイクル、位相の各カウントが PWMxTMR 値にそれぞれ一致する時刻で定義されます。位相カウントが一致するとデューティサイクルが開始し、デューティサイクルカウントが一致するとデューティサイクルが終了します。周期カウントが一致するとタイマがリセットされます。オフセットモードで PWMxTMR を同期させるためにオフセット一致を使う事もできます。詳細は [セクション 26.3「オフセットモード」](#) を参照してください。

標準モードの PWM 周期は [式 26-1](#) で求めます。

標準モードの PWM デューティサイクル比は [式 26-2](#) で求めます。

式 26-1: 標準モードの PWM 周期

$$Period = \frac{(PWMxPR + 1) \cdot Prescale}{PWM_clock}$$

式 26-2: 標準モードの PWM デューティサイクル

$$Duty\ Cycle = \frac{(PWMxDC - PWMxPH)}{PWMxPR + 1}$$

[図 26-4](#) に標準モードの詳細なタイミングチャートを示します。

26.2.2 一致時セットモード

一致時セットモード (MODE = 01) では、位相カウントが PWMxTMR 値と一致すると出力をアクティブにします。PWMxCON レジスタの OUT ビットがクリアされるまで、または PWM モジュールが無効になれるまで出力はアクティブ状態を保ちます。このモードではデューティサイクルカウントは影響を与えません。周期カウントは PWMxTMR の最大値のみを決定します。この値を超えた位相カウントの一致は生じません。

このモードでは PWMxOUT ビットを使って PWM の出力をセットまたはクリアできます。このビットへの書き込みは、ビットを書き込んだ直後の PWM_clock の立ち上がりエッジで実行されます。

[図 26-5](#) に一致時セットの詳細なタイミングチャートを示します。

26.2.3 一致時トグルモード

一致時トグルモード (MODE = 10) では、標準 PWM モードでの計算値の 2 倍の周期を持つデューティサイクル 50% の PWM を生成します。このモードではデューティサイクルカウントは影響を与えません。出力でトグルする周期イベント後の PWMxTMR 周期の数は位相カウントで決まります。

このモードでは PWMxCON レジスタの OUT ビットへの書き込みは影響を与えません。

[図 26-6](#) に一致時トグルの詳細なタイミングチャートを示します。

26.2.4 センターアラインモード

センターアラインモード (MODE = 11) では、周期の中心に位置合わせされた PWM 波形を生成します。このモードでは周期は PWMxPR カウントの 2 倍です。PWMxTMR は周期の値に達するまでカウントアップし、次に 0 までカウントダウンします。デューティサイクルカウントは PWM 出力のアクティブ化の開始と終了の両方を決定します。デューティサイクルは PWMxTMR インクリメント時の一致イベントで開始し、PWMxTMR デクリメント時の一致イベントで終了します。インクリメント時の一致値は、周期カウントからデューティサイクルカウントを引いたものです。デクリメント時の一致値は、インクリメント時の一致値に 1 を加えたものです。

センターアラインモードの PWM 周期は [式 26-3](#) で求めます。

式 26-3: センターアラインモードの PWM 周期

$$Period = \frac{(PWMxPR + 1) \cdot 2 \cdot Prescale}{PWM_clock}$$

センターアラインモードの PWM のデューティサイクル比は [式 26-4](#) で求めます。

式 26-4: センターアラインモードの PWM デューティサイクル

$$Duty\ Cycle = \frac{PWMxDC \cdot 2}{(PWMxPR + 1) \cdot 2}$$

このモードでは PWMxOUT への書き込みは影響を与えません。

[図 26-7](#) にセンターアラインモードの詳細なタイミングチャートを示します。

図 26-4: 標準 PWM モードのタイミング図

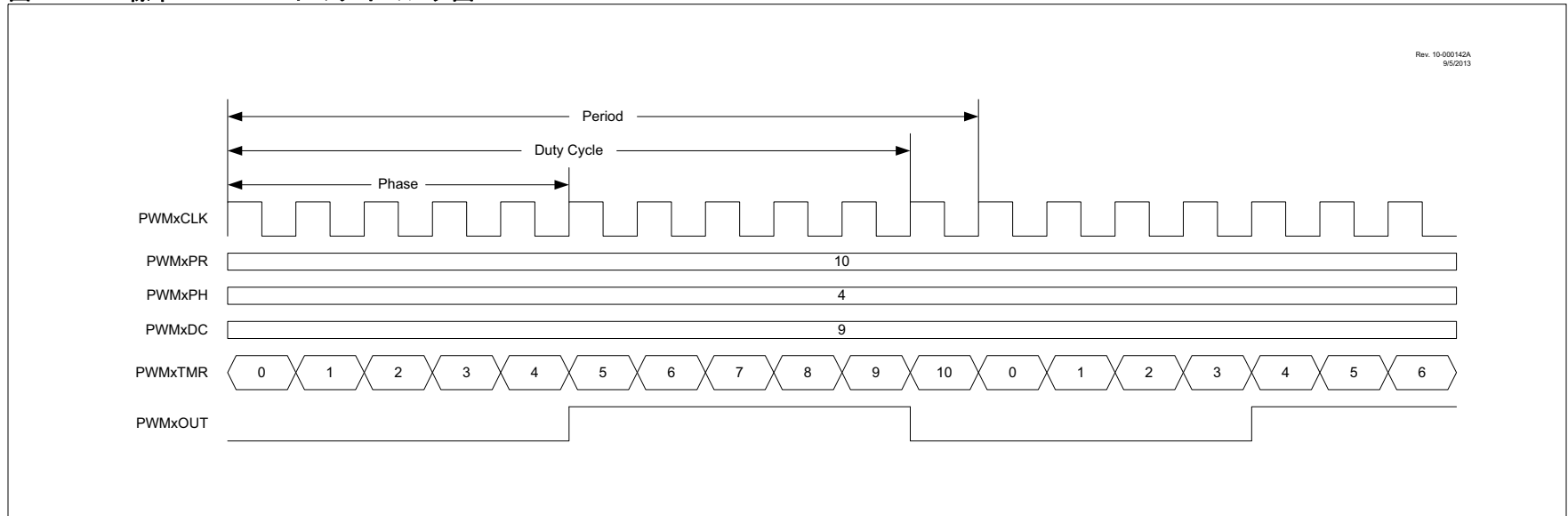
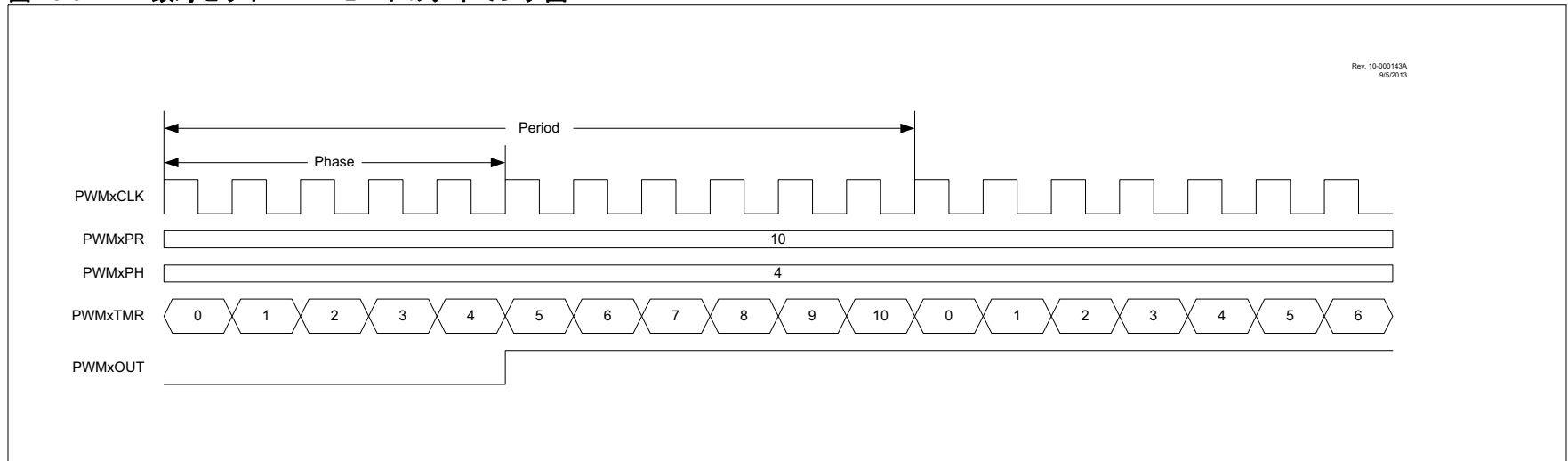
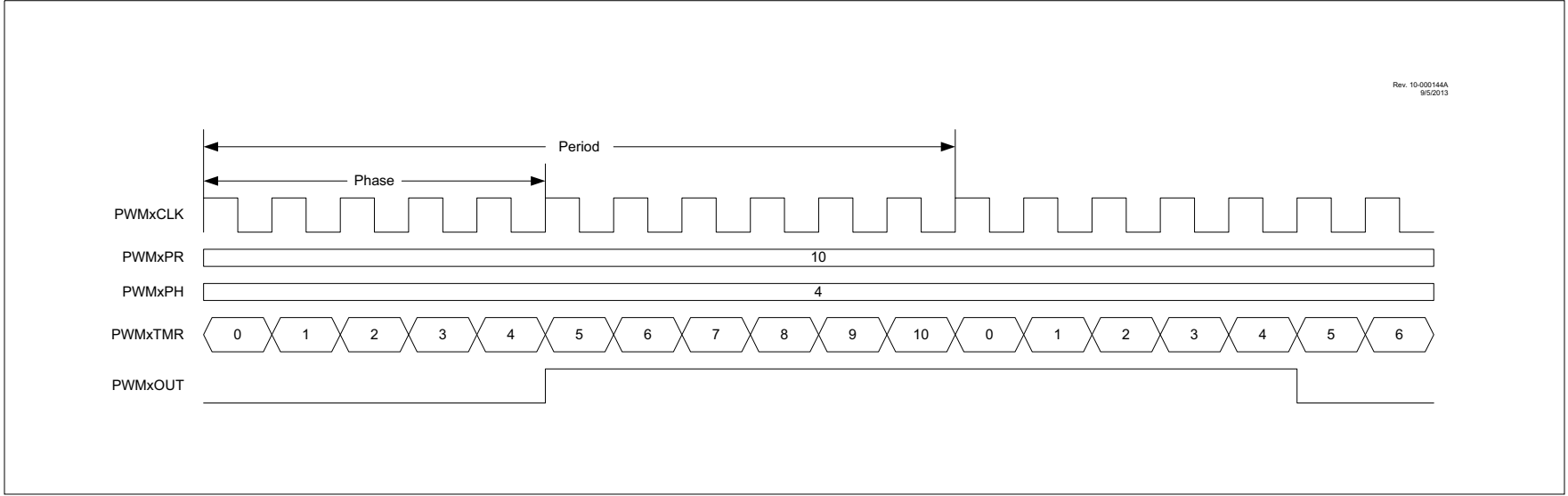


図 26-5: 一致時セット PWM モードのタイミング図



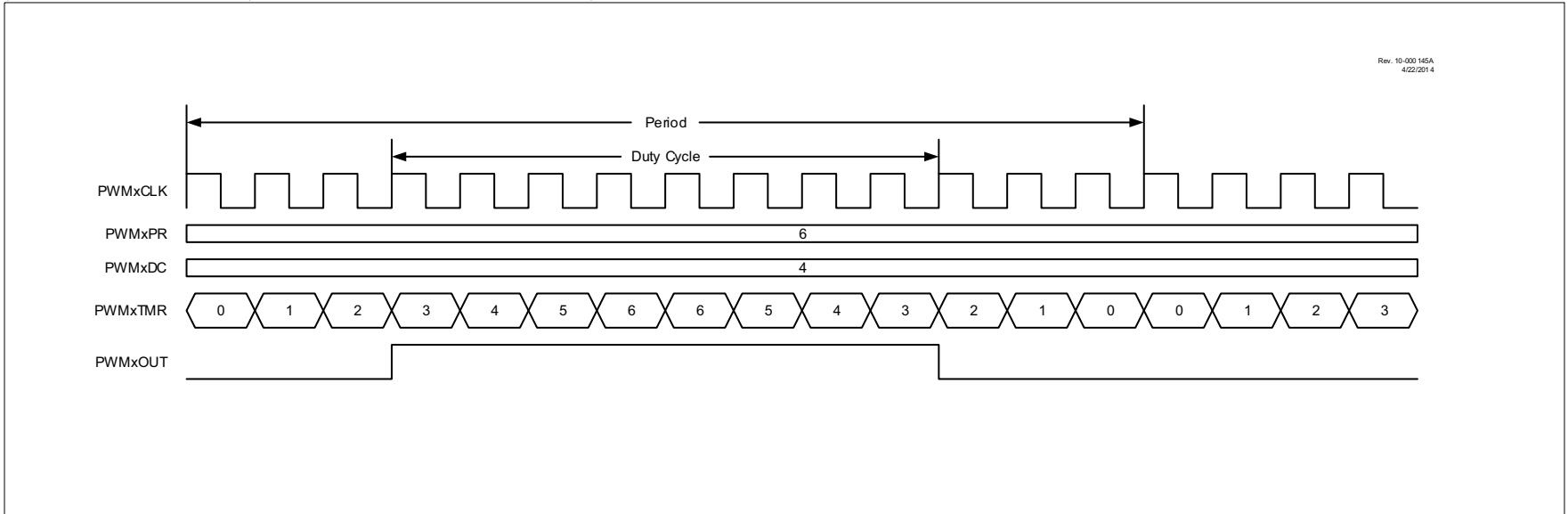
Rev. 10-00144A
9/5/2013

図 26-6: 一致時トグル PWM モードのタイミング図



Rev. 10-000145A
4/22/2014

図 26-7: センターアライン PWM モードのタイミング図



PIC16(L)F1764/5/8/9

26.3 オフセットモード

オフセットモードを使うと、同一デバイス上のマスタ PWM モジュールの波形に対してスレーブ PWM モジュールの波形を調節できます。

26.3.1 独立実行モード

独立実行モード (OFM = 00) では、PWM モジュールは同一デバイス上の他の PWM モジュールの影響を受けません。このモードの PWM モジュールに関連する PWMxTMR は、関連する EN ビットがセットされると即座にカウントを開始し、EN ビットがクリアされるまでカウントし続けます。周期イベントで PWMxTMR はゼロにリセットされ、その後もタイマはカウントを続けます。

図 26-8 に、標準 PWM モードを使ったこのモードの詳細タイミング図を示します。

26.3.2 同期スタート付き実行スレーブモード

同期スタート付き実行スレーブモード (OFM = 01) では、スレーブ PWMxTMR はマスタの OF_match イベントを待機します。このイベントが発生すると、EN ビットがセットされている場合 PWMxTMR はカウントを開始し、ソフトウェアが EN ビットをクリアするまでカウントし続けます。スレーブ周期イベントで PWMxTMR はゼロにリセットされ、その後もタイマはカウントを続けます。

図 26-9 に、標準 PWM モードを使ったこのモードの詳細タイミング図を示します。

26.3.3 同期スタート付きワンショットスレーブモード

同期スタート付きワンショットスレーブモード (OFM = 10) では、スレーブ PWMxTMR はマスタの OF_match イベントを待機します。このイベントが発生すると、タイマはタイマ内の既存の値からカウントを開始し、周期一致イベントまでカウントし続けます。周期一致イベントが発生すると、タイマはリセットされカウントを停止します。タイマは次のマスタ OF_match イベントを待機し、このイベントの後カウントを再開しサイクルを繰り返します。

図 26-10 に標準 PWM モードを使ったこのモードの詳細タイミング図を示します。

26.3.4 同期スタート、タイマリセット付き連続実行スレーブモード

同期スタート、タイマリセット付き連続実行スレーブモード (OFM = 11) では、スレーブ PWM イネーブルをセットした後はスレーブ PWMxTMR はカウントしません。マスタの最初の OF_match イベントでスレーブ PWMxTMR はカウントを開始します。マスタのその後の OF_match イベントでスレーブ PWMxTMR タイマ値は 1 にリセットされ、その後もスレーブ PWMxTMR はカウントを続けます。マスタの次の OF_match イベントでスレーブ PWMxTMR は 1 にリセットされ、サイクルを繰り返します。マスタの OF_match イベントの前にスレーブの周期イベントが発生すると、スレーブ PWMxTMR はゼロにリセットされ、その後もタイマは

カウントを続けます。このモードで動作するスレーブは PWMxPH レジスタペアの値が 1 以上である必要があります。そうでない場合、位相一致イベントは発生せず PWM 出力デューティ サイクルは開始できません。マスタおよびスレーブ PWMxPR の値が同じでかつ PWM モジュールが動作中にスレーブ オフセットモードが独立実行モードに変わった場合、このオフセットタイミングは持続します。

図 26-11 に、標準 PWM モードを使ったこのモードの詳細タイミング図を示します。

Note: スレーブの PWM_clock の周波数がマスタの PWM_clock より大きい場合、予期せぬ結果が生じます。

26.3.5 センターアライン モードでのオフセット一致

マスタがセンターアライン モードで動作中、オフセット一致イベントは PWMxTMR のカウント方向によって異なります。PWMxOFCON レジスタの OFO ビットをクリアすると、タイマがカウントアップしている時に OF_match イベントが発生します。PWMxOFCON レジスタの OFO ビットをセットすると、タイマがカウントダウンしている時に OF_match イベントが発生します。センターアライン モード以外では OFO ビットは無視されます。

OFO ビットはダブルバッファ構成であり、PWM モジュール動作時に有効にするには LDA ビットをセットする必要があります。

図 26-12 と図 26-13 に、同期スタート付き独立スレーブモードのオフセット一致を使ったセンターアラインモードの詳細タイミング図を示します。

図 26-8: 独立実行モードのタイミング図

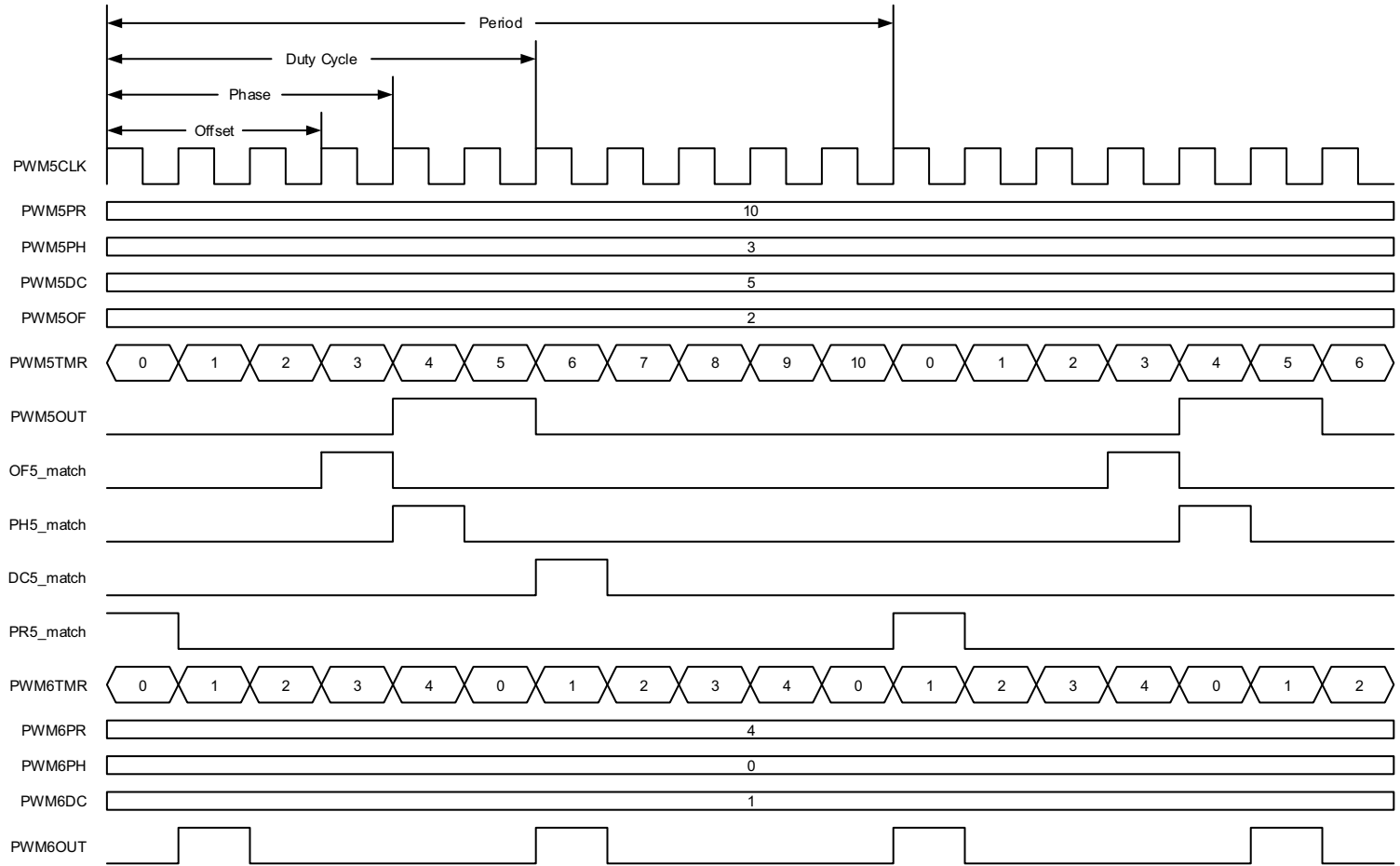
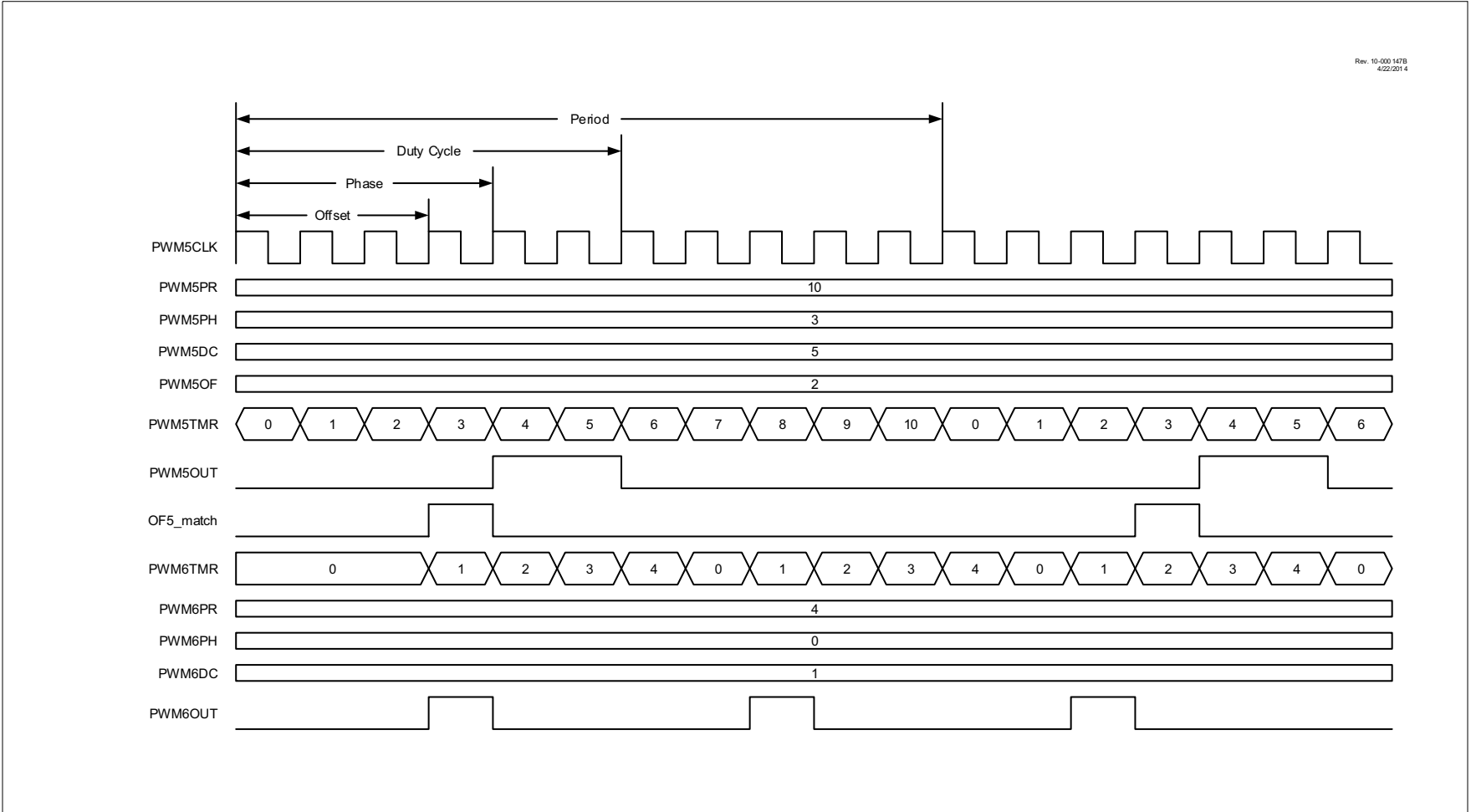


図 26-9: 同期スタート付き実行スレーブモードのタイミング図



Rev. 10-001-148B
4/22/2014

図 26-10: 同期スタート付きワンショット実行スレーブモードのタイミング図

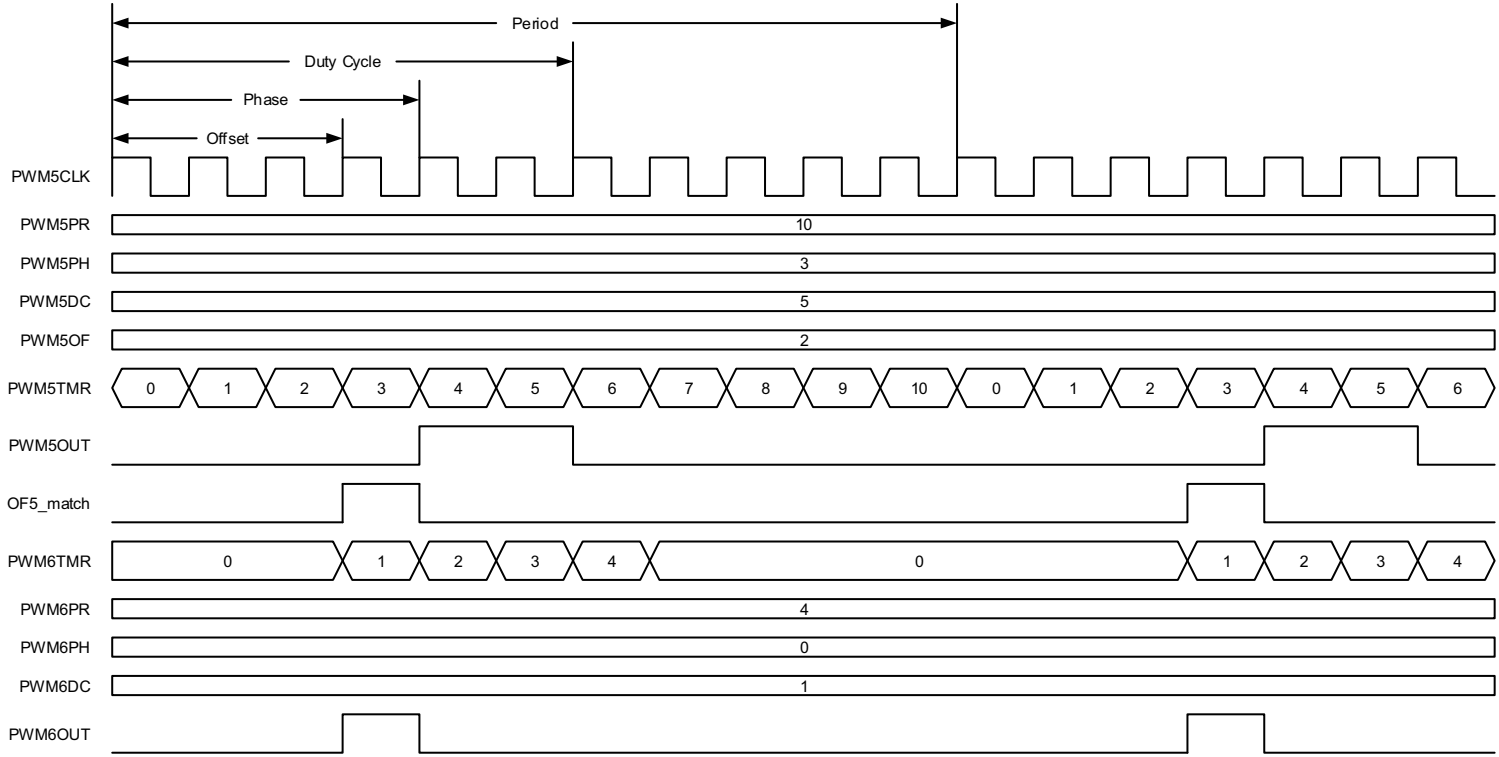


図 26-11: 即座リセット、同期スタート付き連続実行スレーブモードのタイミング図

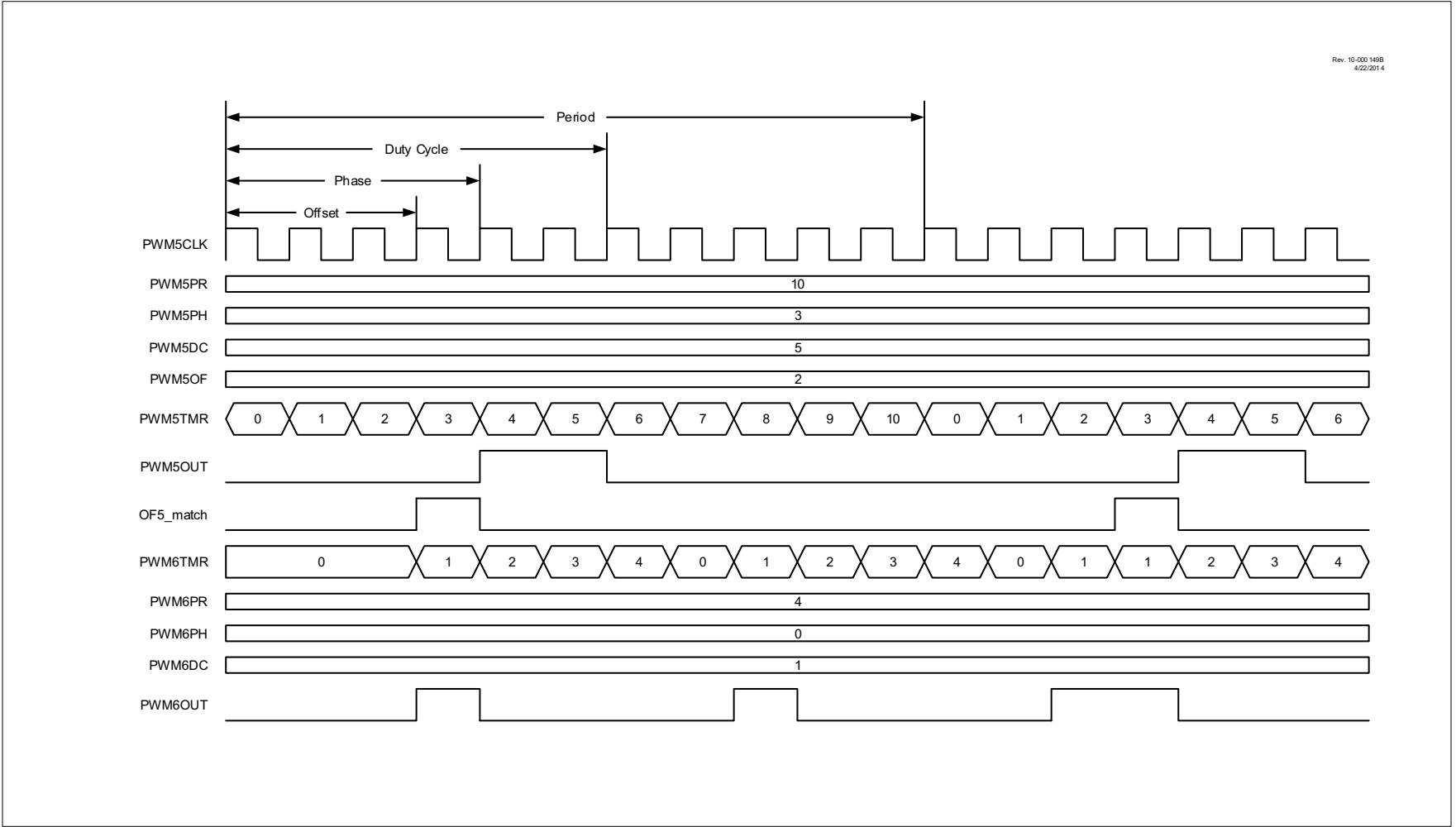


図 26-12: インクリメント タイマオフセット一致のタイミング図

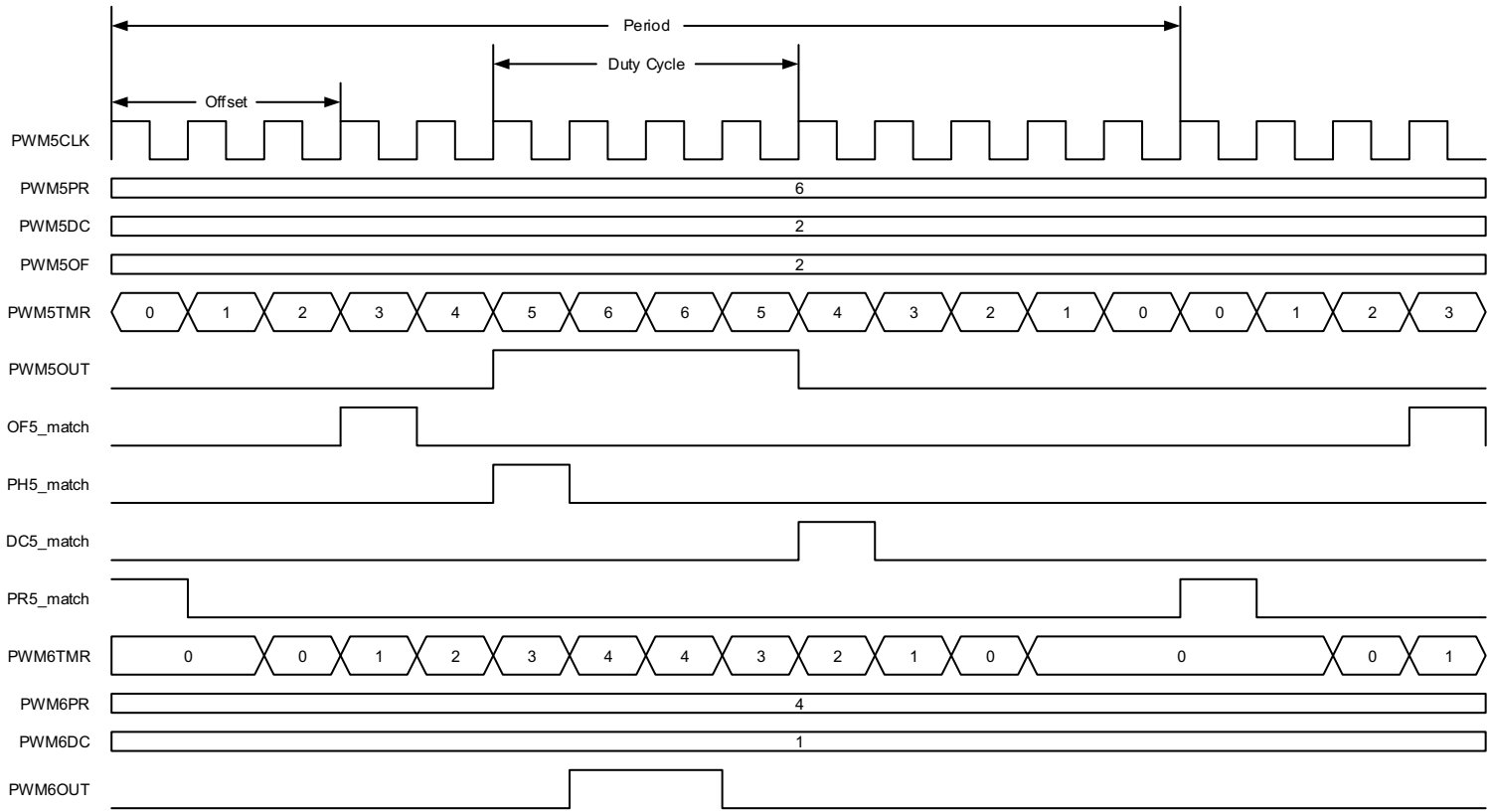
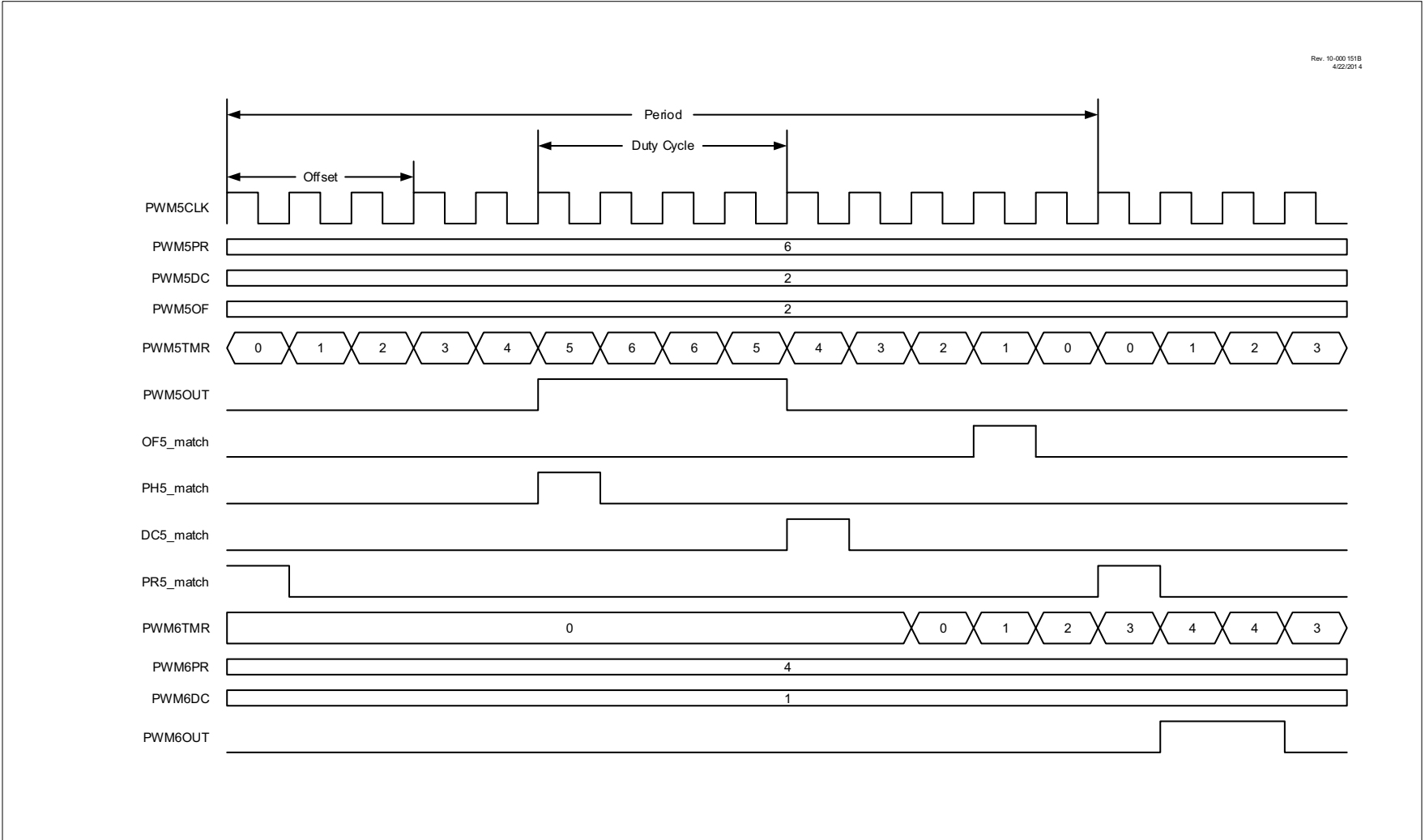


図 26-13: デクリメント タイマオフセット一致のタイミング図



26.4 再書き込み動作

4つのPWMモジュール制御レジスタペアと1つの制御ビットはダブルバッファ構成であるため、全てが同時に更新できます。これには下記を含みます。

- PWMxPHH:PWMxPHL レジスタペア
- PWMxDCH:PWMxDCL レジスタペア
- PWMxPRH:PWMxPRL レジスタペア
- PWMxOFH:PWMxOFL レジスタペア
- ODO 制御ビット

これらのレジスタに書き込んでも即座にはPWMの動作に影響しません。既定値では、これらのレジスタへ書き込んでもアーム条件が満たされない限りPWM動作バッファレジスタには書き込まれません。アーム制御には以下の2種類の動作方法があります。

- 即座
- トリガ

PWMxLDLCON レジスタのLDTビットでアーム方法を制御します。どちらの方法でもLDTビットをセットする必要があります。書き込みイベントで4つのバッファペアは全て同時に書き込まれます。

26.4.1 即座再書き込み

LDTビットをクリアすると即座モードが選択され、バッファはLDAビットがセットされた後最初の周期イベントで書き込まれます。即座再書き込みは、PWMモジュールがスタンダアロンで動作中または他のスレーブPWMモジュールのマスターPWMモジュールとして動作中に使います。

26.4.2 トリガ再書き込み

LDTビットをセットしてトリガモードを選択すると、LDAビットが有効となるためにトリガイイベントが必要です。このトリガ源は、デバイス内の他のPWMモジュールのバッファ書き込みイベントです。トリガ源はPWMxLDLCON レジスタのLDS<1:0>ビットで選択します。トリガイイベントの直後の周期イベントでバッファが書き込まれます。PWMモジュールが他のPWMモジュールのスレーブとして動作中で、かつ両モジュールでのバッファ再書き込みを同期させる事が必要な場合、トリガ再書き込みを使います。

Note 1: バッファの書き込み動作でLDAビットがクリアされます。

2: PWMxTMR = PWMxPR でかつLDAビットがセットされた場合、LDAビットは次の周期イベントまで無視されます。トリガ再書き込みが選択されている時に上記のようなケースが生じた場合、周期イベントと同時にトリガイイベントが発生します。

26.5 スリープ中の動作

PWMモジュールのクロック源としてPWMxCLKCON<1:0>でHFINTOSCまたはLFINTOSCを選択した場合、スリープ中も動作を継続します。

26.6 割り込み

各PWMモジュールは、位相、デューティサイクル、周期、オフセット一致イベントに基づく4種類の独立した割り込みを備えます。割り込みフラグはこれらの信号の立ち上がりエッジで設定されます。一致信号の詳細タイミング図は図26-8および26-12を参照してください。

PIC16(L)F1764/5/8/9

26.7 レジスタ定義 : PWM 制御

表 26-1 に、16 ビット PWM 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1「レジスタおよびビット命名法」](#) を参照してください。

表 26-1:

周辺モジュール	ビット名の接頭辞
PWM5	PWM5
PWM6 ⁽¹⁾	PWM6

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 26-1: PWMxCON: PWM 制御レジスタ

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>		—	—
bit 7							bit 0

凡例:

HC = ビットはハードウェアでクリア

HS = ビットはハードウェアでセット

R = 読み出し可能ビット

W = 書き込み可能ビット

U = 未実装ビット、「0」として読み出し

u = ビットは不変

x = ビットは未知

-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット

「0」= ビットはクリア

- bit 7 **EN:** PWM モジュール イネーブルビット
1 = モジュールを有効にする
0 = モジュールを無効にする
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **OUT:** PWM モジュールの出力状態
- bit 4 **POL:** PWM 出力極性制御ビット
1 = PWM 出力のアクティブ状態を Low にする
0 = PWM 出力のアクティブ状態を High にする
- bit 3-2 **MODE<1:0>:** PWM モード制御ビット
11 = センターアライン モード
10 = 一致時トグルモード
01 = 一致時セットモード
00 = 標準 PWM モード
- bit 1-0 **未実装:** 「0」として読み出し

PIC16(L)F1764/5/8/9

レジスタ 26-2: PWMxINTE: PWM 割り込みイネーブル レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OFIE	PHIE	DCIE	PRIE
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-4 **未実装:** 「0」として読み出し
- bit 3 **OFIE:** オフセット割り込みイネーブルビット
1 = オフセット一致時に CPU に対して割り込む
0 = オフセット一致時に CPU に対して割り込まない
- bit 2 **PHIE:** 位相割り込みイネーブルビット
1 = 位相一致時に CPU に対して割り込む
0 = 位相一致時に CPU に対して割り込まない
- bit 1 **DCIE:** デューティ サイクル割り込みイネーブルビット
1 = デューティ サイクル一致時に CPU に対して割り込む
0 = デューティ サイクル一致時に CPU に対して割り込まない
- bit 0 **PRIE:** 周期割り込みイネーブルビット
1 = 周期一致時に CPU に対して割り込む
0 = 周期一致時に CPU に対して割り込まない

レジスタ 26-3: PWMxINTF: PWM 割り込み要求レジスタ

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	OFIF	PHIF	DCIF	PRIF
bit 7				bit 0			

凡例:

HC = ビットはハードウェアでクリア HS = ビットはハードウェアでセット
R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7-4 **未実装:** 「0」として読み出し
- bit 3 **OFIF:** オフセット割り込みフラグビット⁽¹⁾
1 = オフセット一致イベントが発生した
0 = オフセット一致イベントは発生していない
- bit 2 **PHIF:** 位相割り込みフラグビット⁽¹⁾
1 = 位相一致イベントが発生した
0 = 位相一致イベントは発生していない
- bit 1 **DCIF:** デューティ サイクル割り込みフラグビット⁽¹⁾
1 = デューティ サイクル一致イベントが発生した
0 = デューティ サイクル一致イベントは発生していない
- bit 0 **PRIF:** 周期割り込みフラグビット⁽¹⁾
1 = 周期一致イベントが発生した
0 = 周期一致イベントは発生していない

PIC16(L)F1764/5/8/9

レジスタ 26-3: PWMxINTF: PWM 割り込み要求レジスタ (続き)

Note 1: モジュールが無効化されている間 (EN = 0)、ビットはハードウェアでクリアされます。

レジスタ 26-4: PWMxCLKCON: PWM クロック制御レジスタ

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	PS<2:0>			—	—	CS<1:0>	
bit 7							bit 0

凡例:

R = 読み出し可能ビット

W = 書き込み可能ビット

U = 未実装ビット、「0」として読み出し

u = ビットは不変

x = ビットは未知

-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

「1」= ビットはセット

「0」= ビットはクリア

bit 7 **未実装:** 「0」として読み出し

bit 6-4 **PS<2:0>:** クロック源プリスケラ選択ビット

111 = クロック源を 128 分周する

110 = クロック源を 64 分周する

101 = クロック源を 32 分周する

100 = クロック源を 16 分周する

011 = クロック源を 8 分周する

010 = クロック源を 4 分周する

001 = クロック源を 2 分周する

000 = プリスケラを使わない

bit 3-2 **未実装:** 「0」として読み出し

bit 1-0 **CS<1:0>:** クロック源選択ビット

11 = 予約済み

10 = LFINTOSC (スリープ中も動作を継続する)

01 = HFINTOSC (スリープ中も動作を継続する)

00 = FOSC

レジスタ 26-5: PWMxLDCON: PWM 再書き込みトリガ源選択レジスタ

R/W/HC-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
LDA ⁽¹⁾	LDT ⁽³⁾	—	—	—	—	—	LDS ^(2, 3)
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	HC = ハードウェアでクリア

bit 7 **LDA:** 書き込みバッファアームビット ⁽¹⁾

LDT = 1 の場合:

- 1 = 選択したトリガが発生した周期の末尾で ODO ビットと OFx、PHx、DCx、PRx の各バッファに書き込む
- 0 = バッファに書き込まずに書き込みを完了する

LDT = 0 の場合:

- 1 = 現在の周期末尾で ODO ビットと OFx、PHx、DCx、PRx の各バッファに書き込む
- 0 = バッファに書き込まずに書き込みを完了する

bit 6 **LDT:** トリガ時バッファ書き込みビット ⁽³⁾

- 1 = LDS<1:0> ビットで選択したトリガが発生したら LDA ビットを有効にする
- 0 = 書き込みトリガを無効にする。バッファの書き込みは LDA ビットでのみ制御される

bit 5-1 **未実装:** 「0」として読み出し

bit 0 **LDS:** 書き込みトリガ源選択ビット ^(2, 3)

- 1 = LD6_trigger
- 0 = LD5_trigger

- Note**
- 1: 再書き込み動作の後、このビットはモジュールによってクリアされます。このビットをソフトウェアでクリアして既存のアームイベントを取り消す事もできます。
 - 2: PWM モジュール自身の LDx_trigger に対応する信号源は予約済みです。
 - 3: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 26-6: PWMxOFCON: PWM オフセットトリガ源選択レジスタ

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
—	OFM<1:0> ⁽²⁾	OFO ⁽¹⁾	—	—	—	—	OFS ⁽²⁾
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **未実装:** 「0」として読み出し
- bit 6-5 **OFM<1:0>:** オフセットモード選択ビット⁽²⁾
11 = オフセットトリガ タイマリセット、同期スタート付き連続実行スレーブモード
10 = オフセットトリガ、同期スタート付きワンショット スレーブモード
01 = オフセットトリガ、同期スタート付き実行スレーブモード
00 = 独立実行モード
- bit 4 **OFO:** オフセット一致出力制御ビット⁽²⁾
MODE<1:0> = 11 (PWM センターアライン モード) の場合:
1 = PWMxTMR カウントアップ中に OFx_match を発生する
0 = PWMxTMR カウントダウン中に OFx_match を発生する
MODE<1:0> = 00、01、10 のいずれかの場合 (その他のモード):
このビットを無視する
- bit 3-1 **未実装:** 「0」として読み出し
- bit 0 **OFS:** オフセットトリガ源選択ビット⁽²⁾
1 = OF6_match
0 = OF5_match

- Note 1:** PWM モジュール自身の OFx_match に対応する信号源は予約済みです。
2: PIC16(L)F1768/9 のみです。

レジスタ 26-7: PWMxPHH: PWMx 位相カウント上位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PH<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **PH<15:8>**: PWM 位相上位ビット
PWM 位相カウントの上位 8 ビット

レジスタ 26-8: PWMxPHL: PWMx 位相カウント下位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PH<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **PH<7:0>**: PWM 位相下位ビット
PWM 位相カウントの下位 8 ビット

PIC16(L)F1764/5/8/9

レジスタ 26-9: PWMxDCH: PWMx デューティ サイクルカウント上位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
DC<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **DC<15:8>**: PWM デューティ サイクル上位ビット
PWM デューティ サイクルカウントの上位 8 ビット

レジスタ 26-10: PWMxDCL: PWMx デューティ サイクルカウント下位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
DC<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **DC<7:0>**: PWM デューティ サイクル下位ビット
PWM デューティ サイクルカウントの下位 8 ビット

レジスタ 26-11: PWMxPRH: PWMx 周期カウント上位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PR<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **PR<15:8>**: PWM 周期上位ビット
PWM 周期カウントの上位 8 ビット

レジスタ 26-12: PWMxPRL: PWMx 周期カウント下位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PR<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **PR<7:0>**: PWM 周期下位ビット
PWM 周期カウントの下位 8 ビット

PIC16(L)F1764/5/8/9

レジスタ 26-13: PWMxOFH: PWMx オフセット カウント上位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
OF<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **OF<15:8>**: PWM オフセット上位ビット
PWM オフセット カウントの上位 8 ビット

レジスタ 26-14: PWMxOFL: PWMx オフセット カウント下位レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
OF<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-0 **OF<7:0>**: PWM オフセット下位ビット
PWM オフセット カウントの下位 8 ビット

レジスタ 26-15: PWMxTMRH: PWMx タイマ上位レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR<15:8>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **TMR<15:8>**: PWM タイマ上位ビット
PWM タイマカウントの上位 8 ビット

レジスタ 26-16: PWMxTMRL: PWMx タイマ下位レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-0 **TMR<7:0>**: PWM タイマ下位ビット
PWM タイマカウントの下位 8 ビット

PIC16(L)F1764/5/8/9

Note: 以下の3つのミラーレジスタについては完全ビット名、短縮ビット名はありません。

レジスタ 26-17: PWMEN: PWMEN ビット ミラーレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6EN ⁽¹⁾	MPWM5EN	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
「1」= ビットはセット 「0」= ビットはクリア -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

bit 7-6 **未実装:** 「0」として読み出し
bit 5-4 **PWMxEN:** PWM6/PWM5 イネーブルビット
 各 PWM モジュールの PWMxCON<7> ビットのミラーコピー
bit 3-0 **未実装:** 「0」として読み出し

Note 1: PIC16(L)F1768/9 のみです (レジスタ 26-18 とレジスタ 26-19 にも適用)。

レジスタ 26-18: PWMLD: LDA ビット ミラーレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6LD ⁽¹⁾	MPWM5LD	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
「1」= ビットはセット 「0」= ビットはクリア -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

bit 7-6 **未実装:** 「0」として読み出し
bit 5-4 **MPWMxLDA:** PWM6/PWM5 LDA ビット
 各 PWM モジュールの PWMxLDLCON<7> ビットのミラーコピー
bit 3-0 **未実装:** 「0」として読み出し

レジスタ 26-19: PWMOUT: PWMOUT ビット ミラーレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6OUT ⁽¹⁾	MPWM5OUT	—	—	—	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
「1」= ビットはセット 「0」= ビットはクリア -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値

bit 7-3 **未実装:** 「0」として読み出し
bit 5-4 **MPWMxOUT:** PWM6/PWM5 OUT ビット
 各 PWM モジュールの PWMxCON<5> ビットのミラーコピー
bit 3-0 **未実装:** 「0」として読み出し

PIC16(L)F1764/5/8/9

表 26-2: PWM モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ	
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		86	
PWMEN	—	—	MPWM6EN ⁽¹⁾	MPWM5EN	—	—	—	—	294	
PWMLD	—	—	MPWM6LD ⁽¹⁾	MPWM5LD	—	—	—	—	294	
PWMOUT	—	—	MPWM6OUT ⁽¹⁾	MPWM5OUT	—	—	—	—	294	
PWM5PHL	PH<7:0>								289	
PWM5PHH	PH<15:8>								289	
PWM5DCL	DC<7:0>								290	
PWM5DCH	DC<15:8>								290	
PWM5PRH	PR<15:8>								291	
PWM5PRL	PR<7:0>								291	
PWM5OFH	OF<15:8>								292	
PWM5OFL	OF<7:0>								292	
PWM5TMRH	TMR<15:8>								293	
PWM5TMRL	TMR<7:0>								293	
PWM5CON	EN	—	OUT	POL	MODE<1:0>		—	—	284	
PWM5INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	285	
PWM5INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	285	
PWM5CLKCON	—	PS<2:0>				—	—	CS<1:0>		286
PWM5LDCON	LDA	LDT ⁽¹⁾	—	—	—	—	—	LDS ⁽¹⁾	287	
PWM5OFCON	—	OFM<1:0> ⁽¹⁾		OFO	—	—	—	OFS ⁽¹⁾	288	
PWM6PHL ⁽¹⁾	PH<7:0>								289	
PWM6PHH ⁽¹⁾	PH<15:8>								289	
PWM6DCL ⁽¹⁾	DC<7:0>								290	
PWM6DCH ⁽¹⁾	DC<15:8>								290	
PWM6PRL ⁽¹⁾	PR<7:0>								291	
PWM6PRH ⁽¹⁾	PR<15:8>								291	
PWM6OFL ⁽¹⁾	OF<7:0>								292	
PWM6OFH ⁽¹⁾	OF<15:8>								292	
PWM6TMRL ⁽¹⁾	TMR<7:0>								293	
PWM6TMRH ⁽¹⁾	TMR<15:8>								293	
PWM6CON ⁽¹⁾	EN	—	OUT	POL	MODE<1:0>		—	—	284	
PWM6INTE ⁽¹⁾	—	—	—	—	OFIE	PHIE	DCIE	PRIE	285	
PWM6INTF ⁽¹⁾	—	—	—	—	OFIF	PHIF	DCIF	PRIF	285	
PWM6CLKCON ⁽¹⁾	—	PS<2:0>				—	—	CS<1:0>		286
PWM6LDCON ⁽¹⁾	LDA	LDT ⁽¹⁾	—	—	—	—	—	LDS ⁽¹⁾	287	
PWM6OFCON ⁽¹⁾	—	OFM<1:0> ⁽¹⁾		OFO	—	—	—	OFS ⁽¹⁾	288	

凡例: — = 未実装、「0」として読み出し。網掛けの部分はPWMでは使いません。

Note 1: PIC16(L)F1768/9のみです。

表 26-3: クロック源関連コンフィグレーションワードのまとめ

レジスタ名	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	レジスタ 内容記載 ページ
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

凡例: — = 未実装、「0」として読み出し。網掛け部分はクロック源では使用しません。

27.0 相補出力ジェネレータ (COG) モジュール

相補出力ジェネレータ (COG) の主な目的は、1 出力の PWM 信号を 2 出力の相補 PWM 信号に変換する事です。また、2 つの別々の入力イベントを単相または相補 PWM 出力に変換する事もできます。

COG の PWM 周波数とデューティ サイクルは、立ち上がりイベント入力と立ち下がりイベント入力が決まります。立ち上がりイベントと立ち下がりイベントに同じ信号源を使う事もできます。入力は、COG_clock に対する同期信号でも非同期信号でもかまいません。

PWM 周波数は、立ち上がりイベントの発生レートで決まります。デューティ サイクルは、立ち上がりイベントから立ち下がりイベントまでの時間で決まります。

位相遅延、ブランキング、デッドバンド遅延の生成には、選択可能なクロック入力を使います。デッドバンド遅延は、全てのクロック源から独立したプログラマブルな非同期遅延チェーンとする事もできます。

図 27-2 ~ 図 27-6 に、各種 COG モードの概略ブロック図を示します。

COG モジュールには以下の特長があります。

- 6 つの動作モード：
 - PWM ステアリング モード
 - 同期 PWM ステアリング モード
 - 順方向フルブリッジ モード
 - 逆方向フルブリッジ モード
 - ハーフブリッジ モード
 - プッシュプル モード
- COG_clock のクロック源を選択可能
- 立ち上がりイベント入力を個別に選択可能
- 立ち下がりイベント入力を個別に選択可能
- イベントのエッジセンスまたはレベルセンスを個別に選択可能
- 出力極性を個別に選択可能
- 立ち上がりイベントと立ち下がりイベントで個別に長さを設定可能な位相遅延
- 以下の機能を備えたデッドバンド制御：
 - 立ち上がりイベントと立ち下がりイベントで個別に長さを設定可能
 - 同期および非同期タイミング
- 立ち上がりイベントと立ち下がりイベントで個別に長さを設定可能なブランキング制御
- 以下の機能を備えた自動シャットダウン制御：
 - シャットダウン要因を個別に選択可能
 - 自動再起動イネーブル
 - 自動シャットダウン ピン オーバーライド制御 (High、Low、OFF、ハイ インピーダンス)

27.1 ピンへの出力 (全モード共通)

COG モジュールは以下の 4 つの出力を備えます。COGA、COGB、COGC、COGD

各出力で利用可能な波形は、COGxCON0 レジスタの MD<2:0> ビットで選択する動作モードで決まります。各ピンの周辺モジュール信号源制御で、出力に用いる 1 本または複数本のピンを選択します。詳細は RxyPPS レジスタ (レジスタ 12-2) を参照してください。

27.2 イベント駆動型 PWM (全モード共通)

1 つの PWM 入力から PWM および相補出力を生成する以外に、COG は周期的な立ち上がりイベントとそれとは別の立ち下がりイベントから PWM 波形を生成する事もできます。この場合、通常は外付け PWM ドライバ回路内のアナログ帰還信号を立ち下がりイベントの入力として使います。この構成では、スイッチングに起因する過渡サージによって立ち下がりイベントが誤ってトリガされる事があるため、このような立ち下がりイベントをマスクする必要があります。COG は、立ち上がり (立ち下がり) イベントによって出力が駆動されてから一定期間は立ち下がり (立ち上がり) イベント入力をマスクするように設定できます。これを入力ブランキングと呼び、セクション 27.8 「ブランキング制御」で説明します。

場合によっては、外部回路に障害が発生しても保護できる機能が必要です。この場合、フォルト条件によって損傷が生じる前にアクティブな駆動を中止する必要があります。この動作を自動シャットダウンと呼び、セクション 27.10 「自動シャットダウン制御」で説明します。

COG は、別の PWM に対して位相を遅らせて動作するように設定できます。この場合、立ち上がりイベントの発生後、位相遅延タイマで設定した遅延の経過後にアクティブな駆動サイクルが開始します。位相遅延の詳細は、セクション 27.9 「位相遅延」で説明します。

図 27-10 に、1 つの CCP1 入力から生成した、位相遅延とデッドバンドを伴う代表的な動作波形を示します。

27.3 動作モード

27.3.1 PWM ステアリング モード

PWM ステアリング モードでは、イベント入力から生成した PWM 信号が単相 PWM として出力され、4 つの COG 出力の任意の組み合わせに対してステアリングできます。出力のステアリングは、COGxSTR レジスタへの書き込み直後の命令サイクルで有効になります。

同期 PWM ステアリング モードも PWM ステアリング モードと同じですが、出力ステアリングの変更が COGxSTR レジスタへの書き込み後の最初の立ち上がりイベントで有効になる点のみ異なります。固定出力データは同期しません。

図 27-2 と 図 27-3 に、ステアリング モードの構成を示します。

PWM ステアリング モードを選択するには COGxCON0 レジスタ (レジスタ 27-1) の MD<2:0> ビットを「000」に設定し、同期 PWM ステアリング モードを選択するには「001」に設定します。

27.3.2 フルブリッジ モード

順方向および逆方向フルブリッジ モードでは、4 つの COG 出力のうち 2 つが有効で、残りの 2 つは無効です。2 つの有効な出力のうち、1 つは PWM 入力信号で変調され、もう 1 つは 100% デューティ サイクルで常時 ON です。方向を変えると、デッドバンド遅延の経過後に変調信号が出力されます。この遅延期間を利用してドライバがシャットダウンできるため、直列接続したパワーデバイスの貫通電流を防ぐことができます。

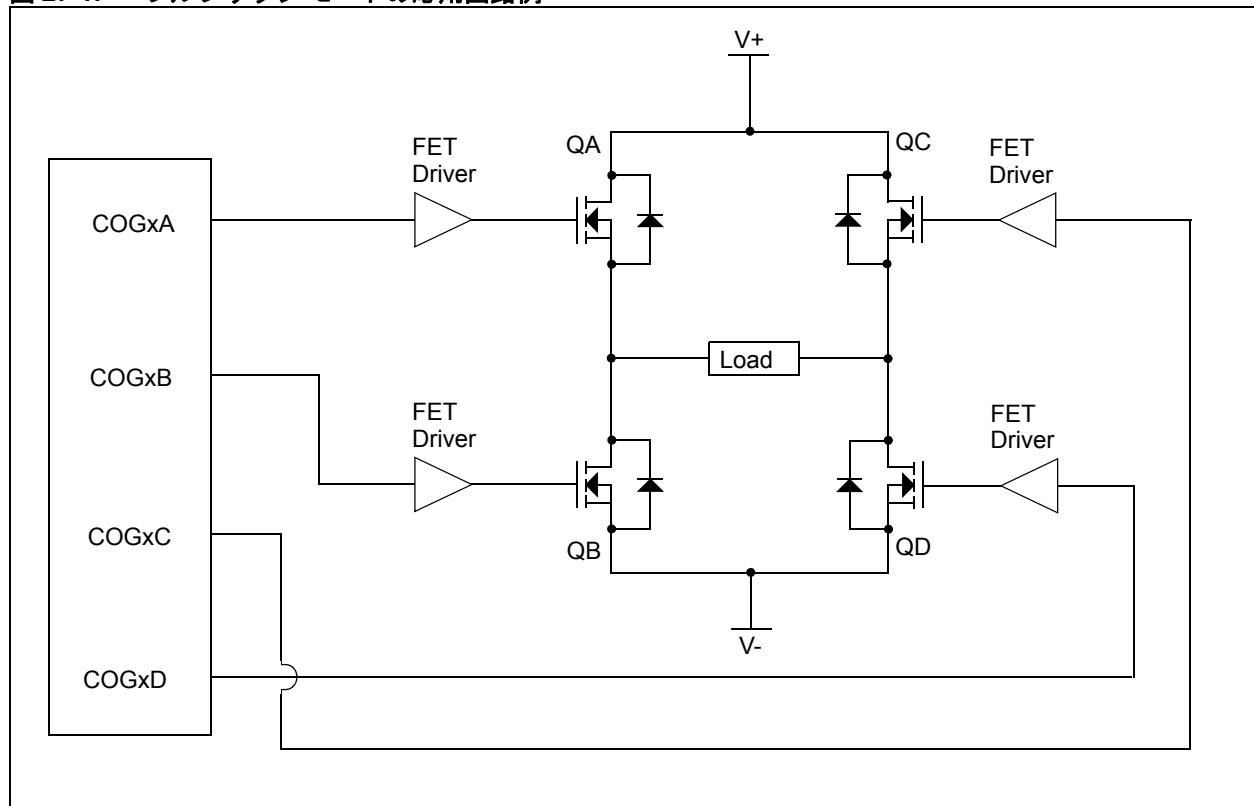
順方向フルブリッジモードでは、PWM 入力がある COGxD 出力を変調し、COGxA 出力を 100% のデューティ サイクルで駆動します。

逆方向フルブリッジモードでは、PWM 入力がある COGxB 出力を変調し、COGxC 出力を 100% のデューティ サイクルで駆動します。

図 27-4 に、フルブリッジ構成を示します。図 27-12 と 図 27-13 に、代表的なフルブリッジ波形を示します。

順方向フルブリッジモードを選択するには COGxCON0 レジスタの MD<2:0> ビットを「010」に設定し、逆方向フルブリッジモードを選択するには「011」に設定します。

図 27-1: フルブリッジモードの応用回路例



PIC16(L)F1764/5/8/9

27.3.3 ハーフブリッジモード

ハーフブリッジモードでは、立ち上がりおよび立ち下がりにイベント入力から1つの2出力相補PWM波形を生成します。最も単純な構成としては、目的の周期とデューティサイクルを持つ1つのPWM信号を、立ち上がりイベント入力と立ち下がりにイベント入力の両方に使います。COGはこの1つのPWM入力を2出力の相補PWM出力に変換します。この2出力PWM信号の周波数とデューティサイクルは、入力に使う1つのPWM信号の周波数とデューティサイクルに一致します。一方の出力がONからOFFへ遷移した後、遅延時間をはさんでもう一方の出力をOFFからONへ遷移させる事で、PWM信号の遷移直後に両方の出力がOFFになる期間を確保できます。この期間をデッドバンド時間と呼び、[セクション 27.7「デッドバンド制御」](#)で説明します。

[図 27-5](#) に、ハーフブリッジ構成を示します。[図 27-9](#) に、1つのCCP1入力から生成した、デッドバンドを伴う代表的な動作波形を示します。

主出力は、COGxAとCOGxCのどちらか、または両方で利用できます。相補出力は、COGxBとCOGxDのどちらか、または両方で利用できます。

ハーフブリッジモードを選択するには、COGxCON0レジスタのMD<2:0>ビットを「100」に設定します。

27.3.4 プッシュプルモード

プッシュプルモードでは、PWM周期ごとに2組のCOG出力が交互に切り換わる1つのPWM出力を生成します。COGxAとCOGxCは同じ信号です。COGxBとCOGxDは同じ信号です。出力の駆動は立ち上がりイベントで開始し、立ち下がりにイベント入力で終了します。立ち上がりイベントごとに新しい周期が開始し、出力は直前の周期で使われていなかったCOGペアへ切り換わります。

[図 27-6](#) に、プッシュプル構成を示します。[図 27-11](#) に、1つのCCP1入力から生成した代表的なプッシュプル波形を示します。

プッシュプルモードを選択するには、COGxCON0レジスタのMD<2:0>ビットを「101」に設定します。

図 27-2: COG の概略ブロック図 (PWM ステアリング モード、MD = 0)

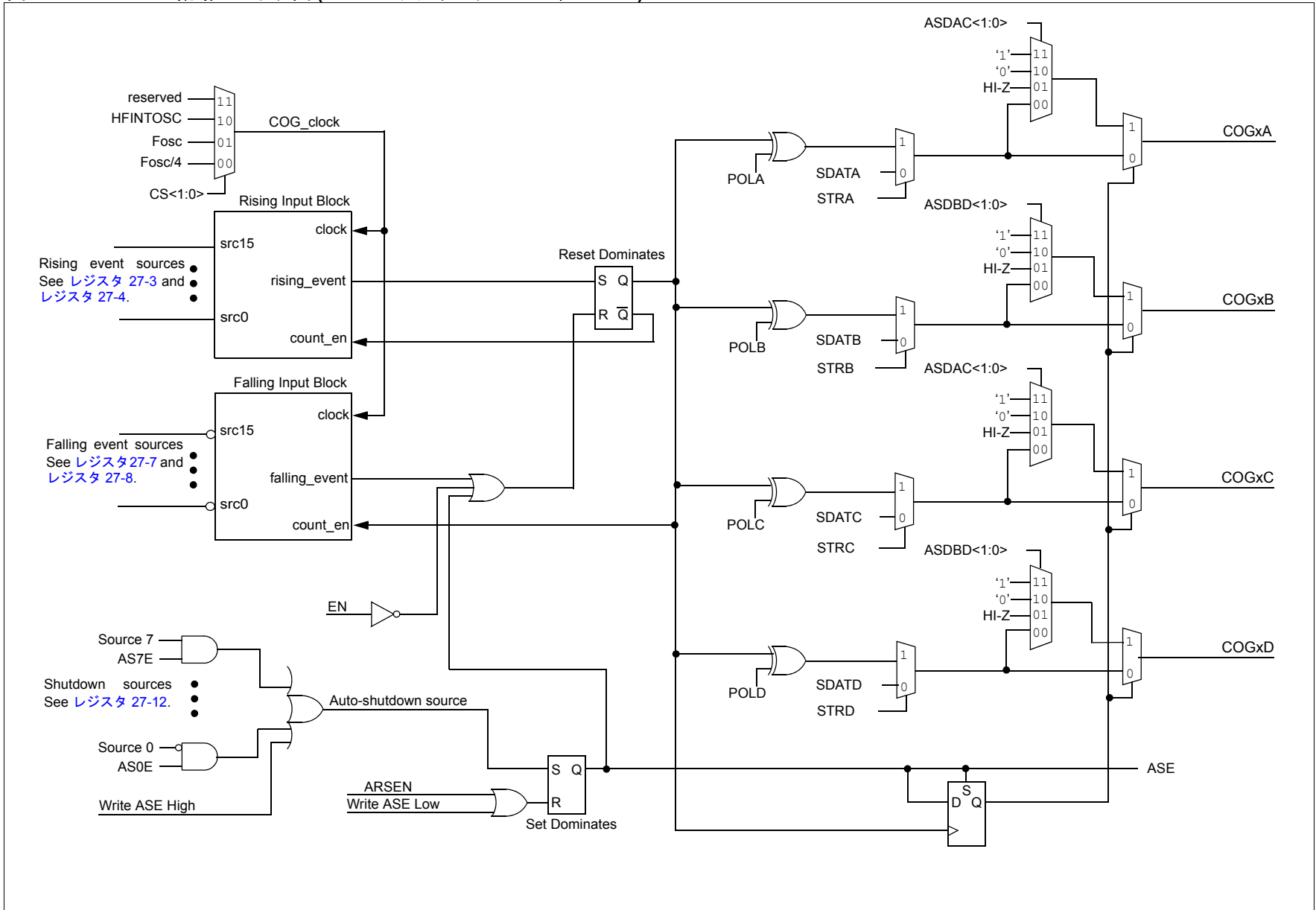


図 27-3: COG の概略ブロック図 (同期 PWM ステアリングモード、MD = 1)

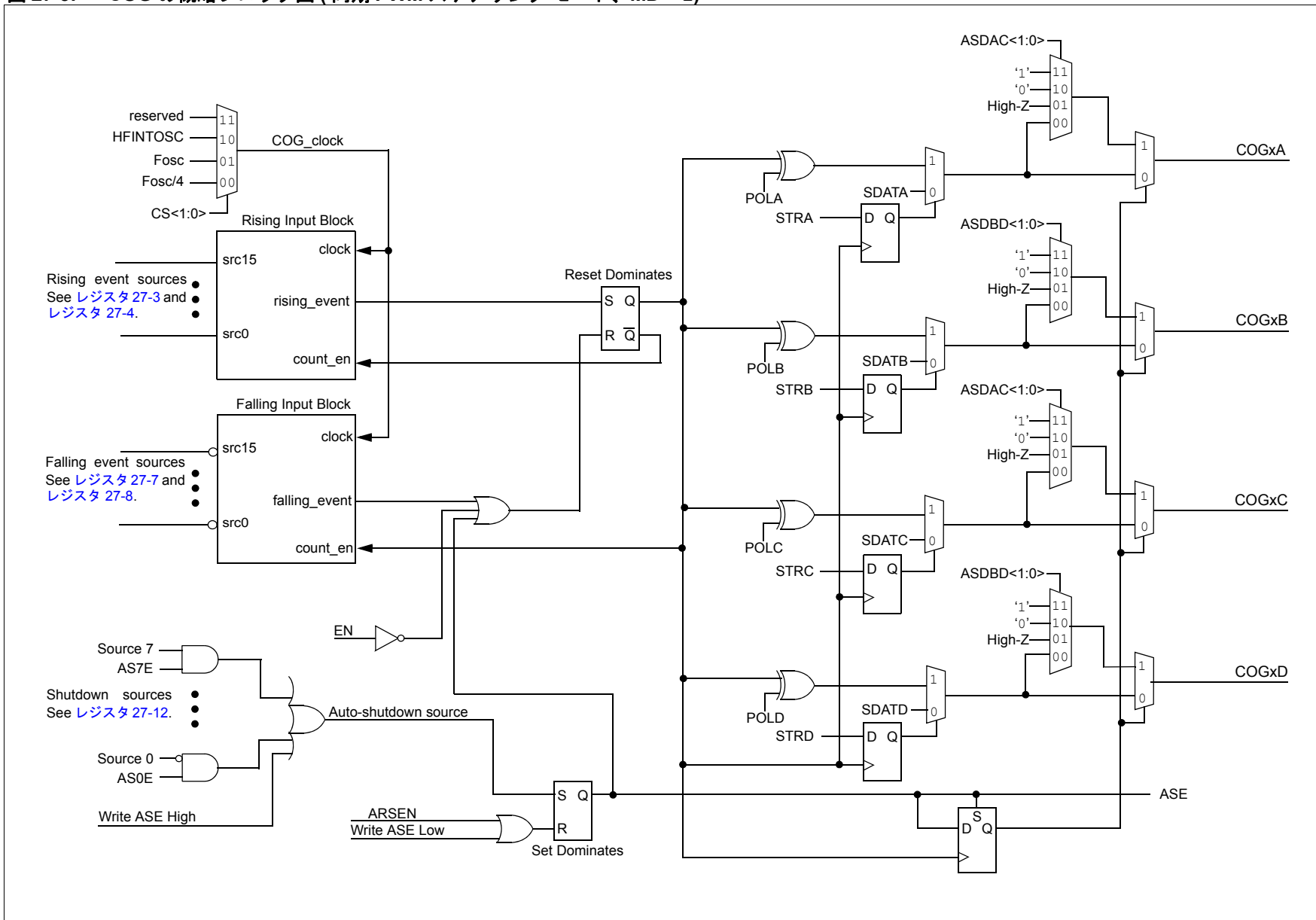


図 27-4: COG の概略ブロック図 (フルブリッジモード、順方向: MD = 2、逆方向: MD = 3)

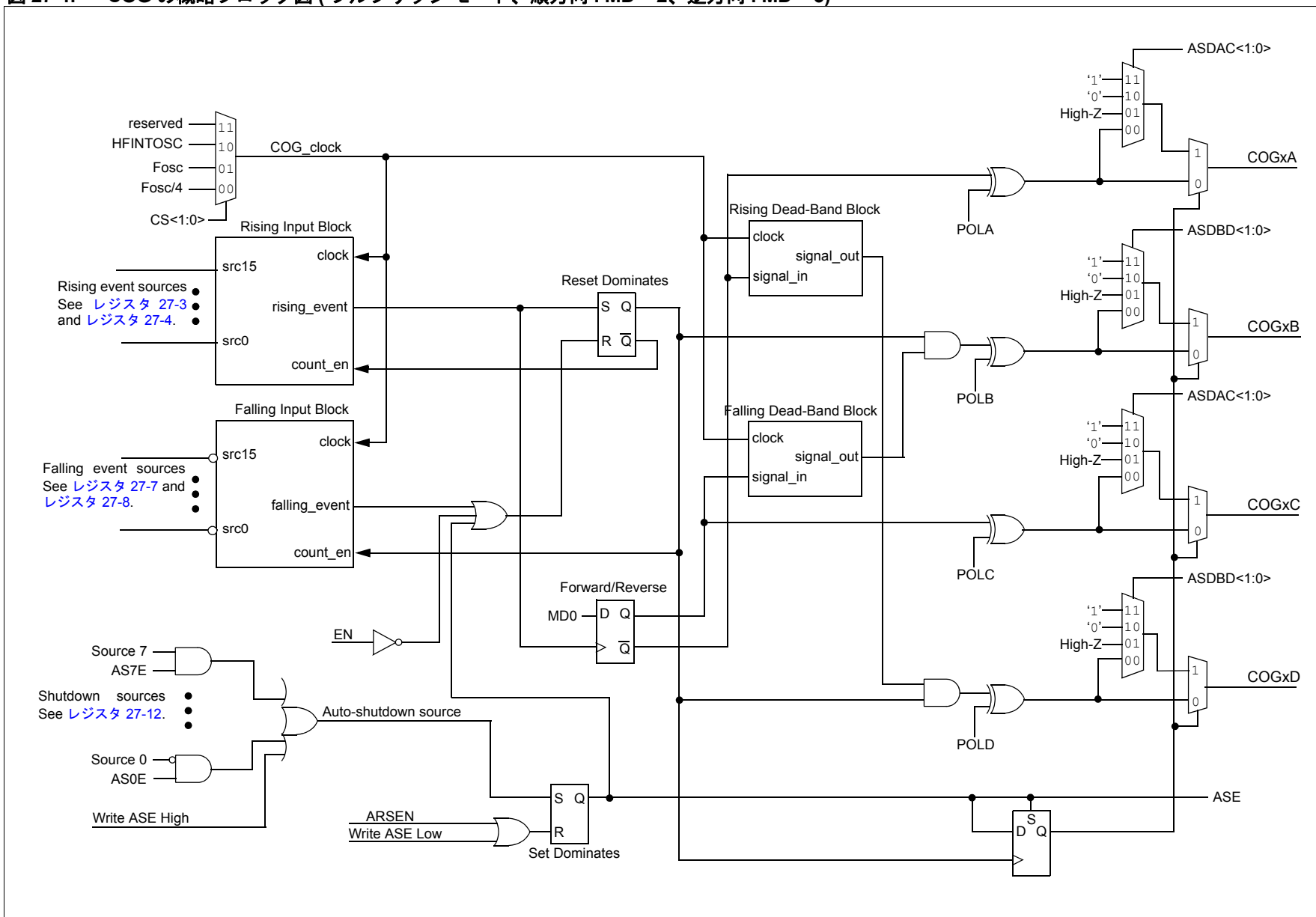


図 27-5: COG の概略ブロック図 (ハーフブリッジモード、MD = 4)

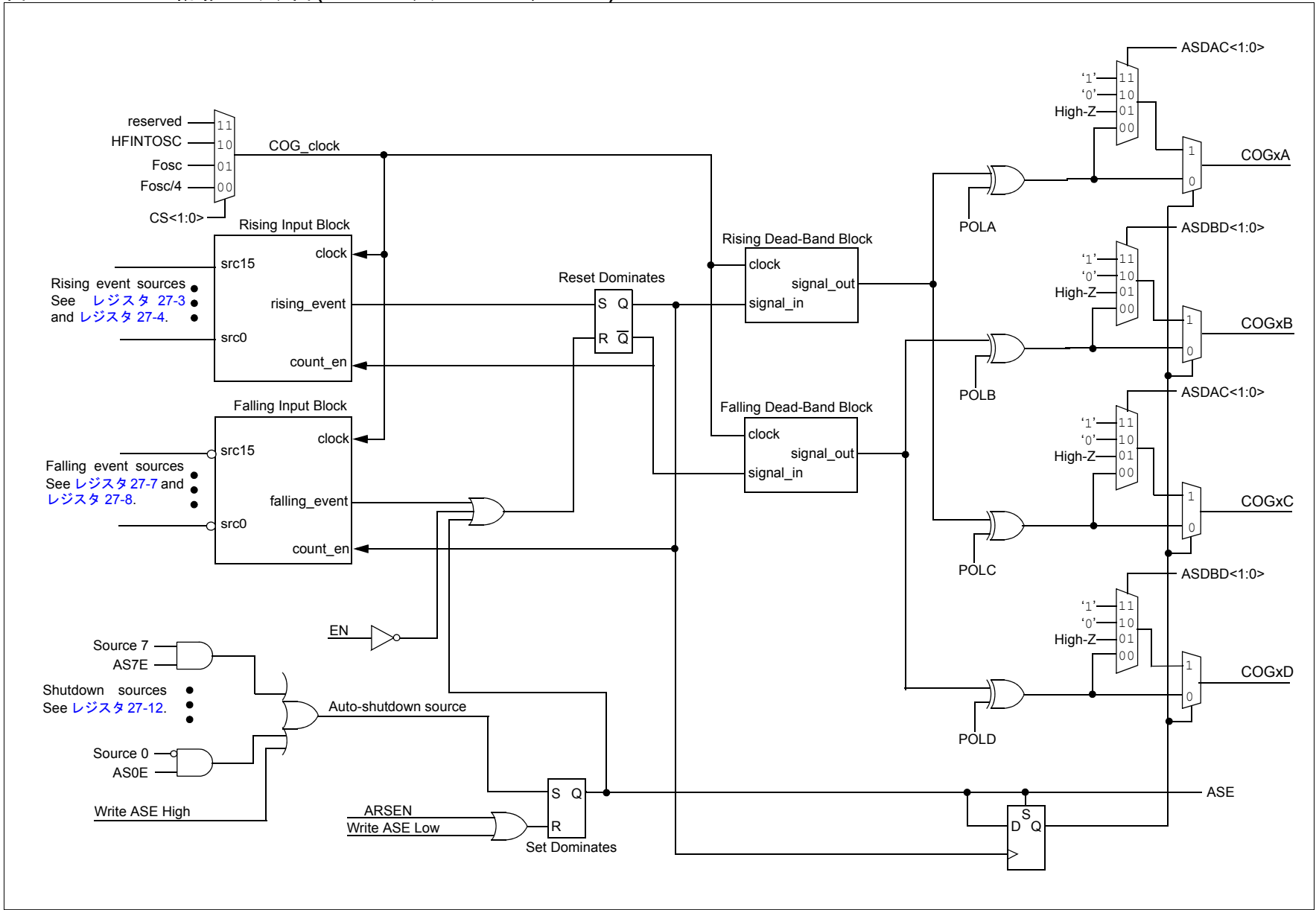
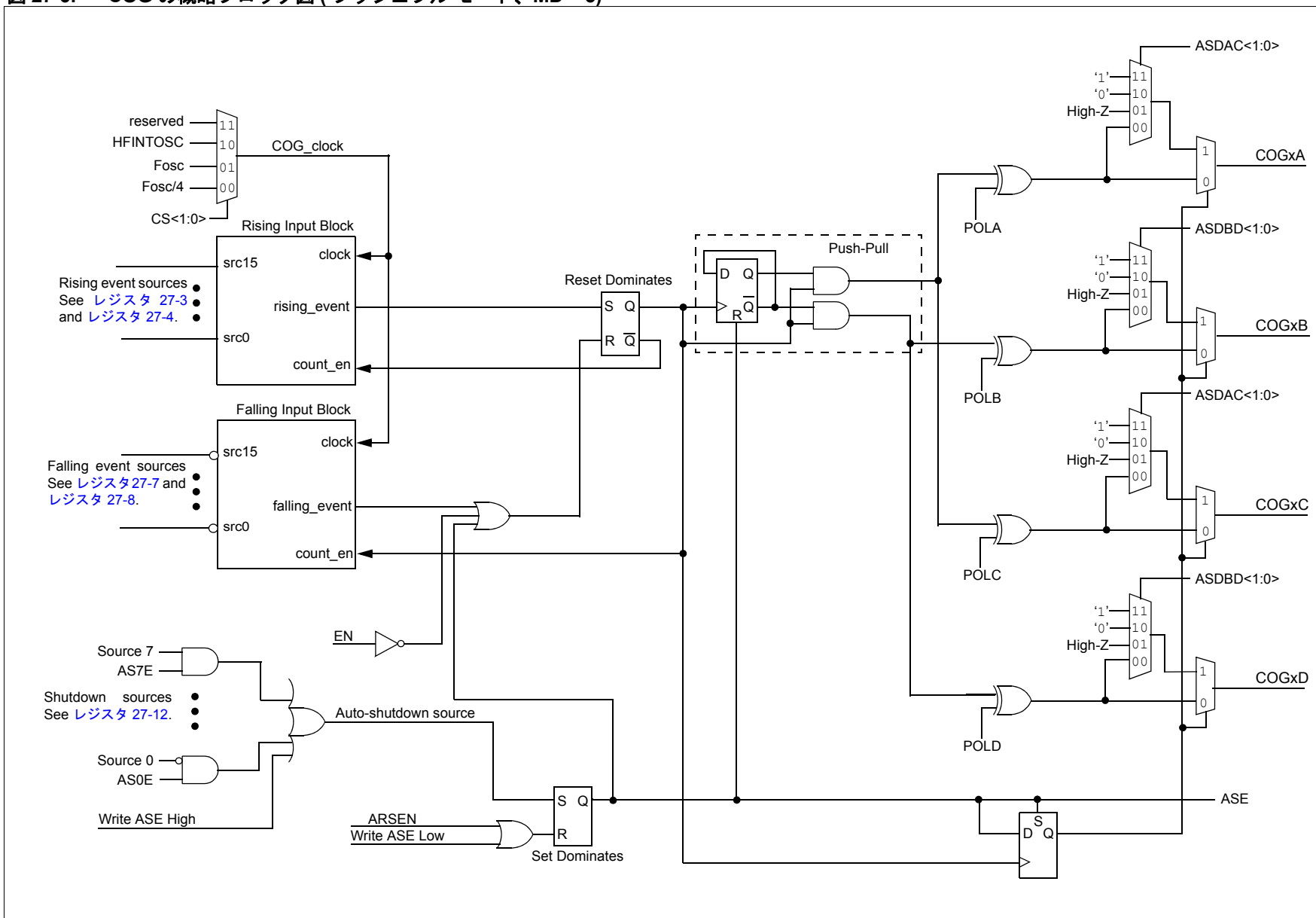


図 27-6: COG の概略ブロック図 (プッシュプルモード、MD = 5)



PIC16(L)F1764/5/8/9

図 27-7: COG(立ち上がり / 立ち下がり) 入力ブロック

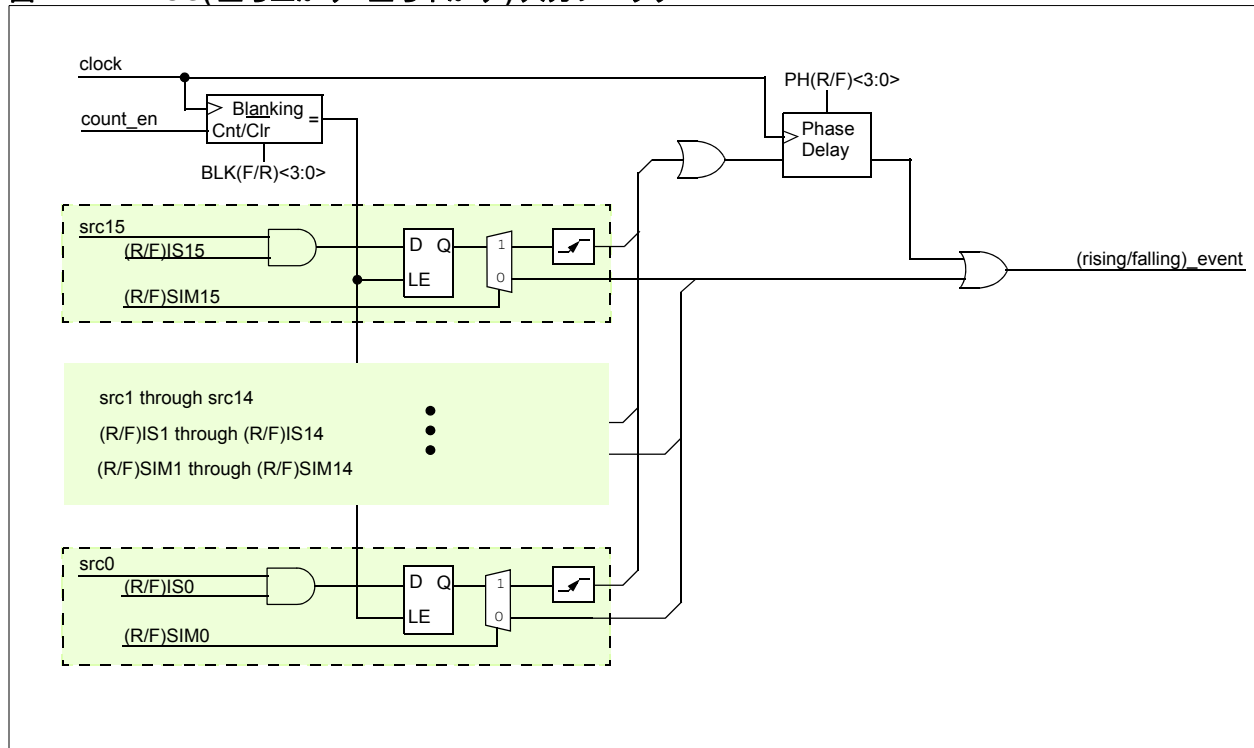


図 27-8: COG(立ち上がり / 立ち下がり) デッドバンドのブロック図

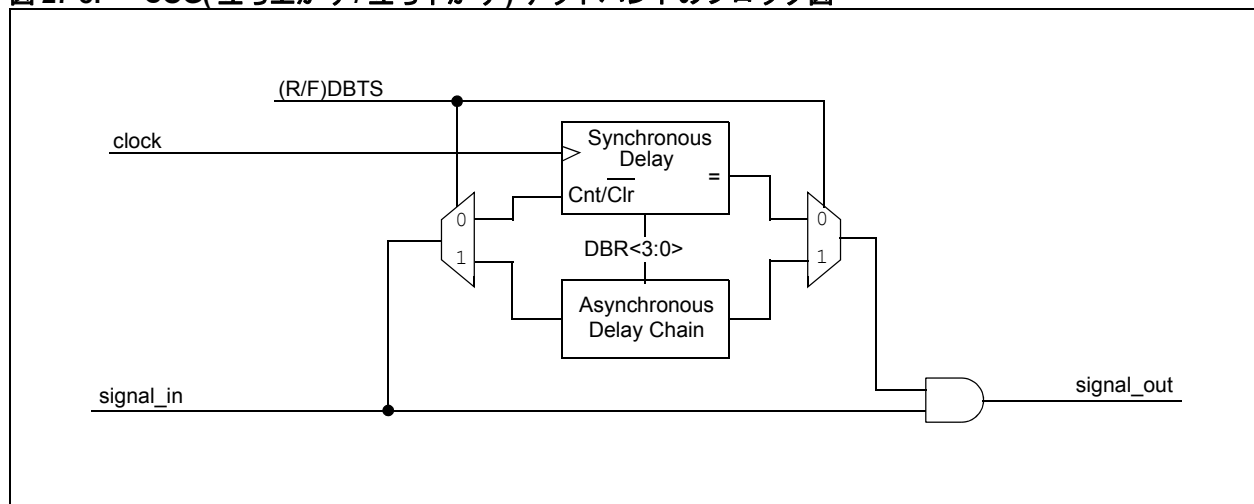


図 27-9: CCP1 を使った代表的なハーフブリッジモードの COG の動作

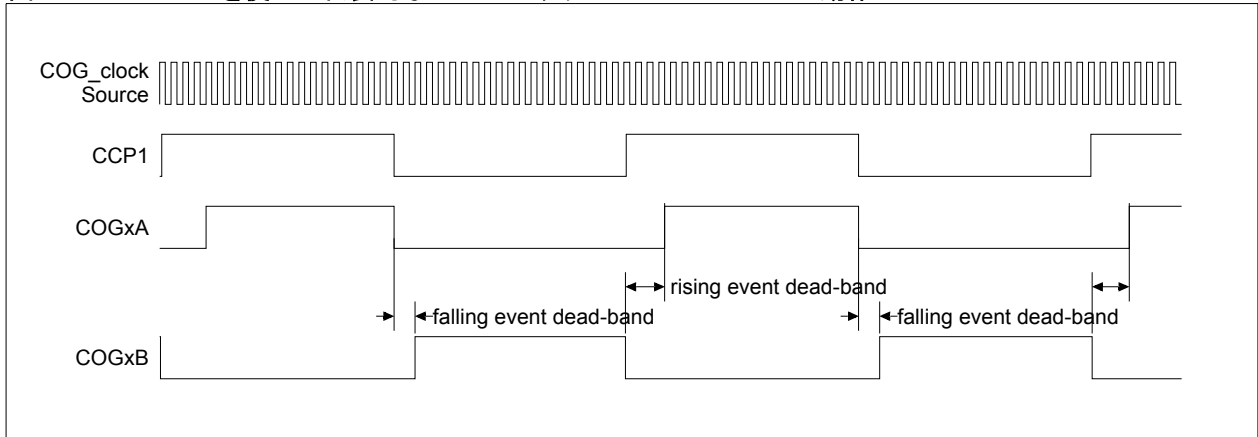


図 27-10: CCP1 を使ったハーフブリッジモードの COG の動作 (位相遅延あり)

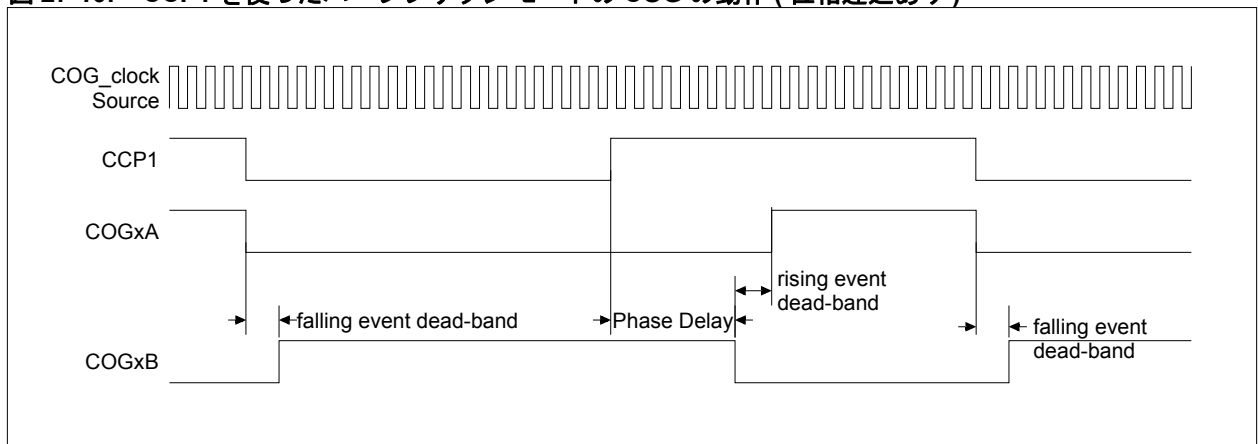
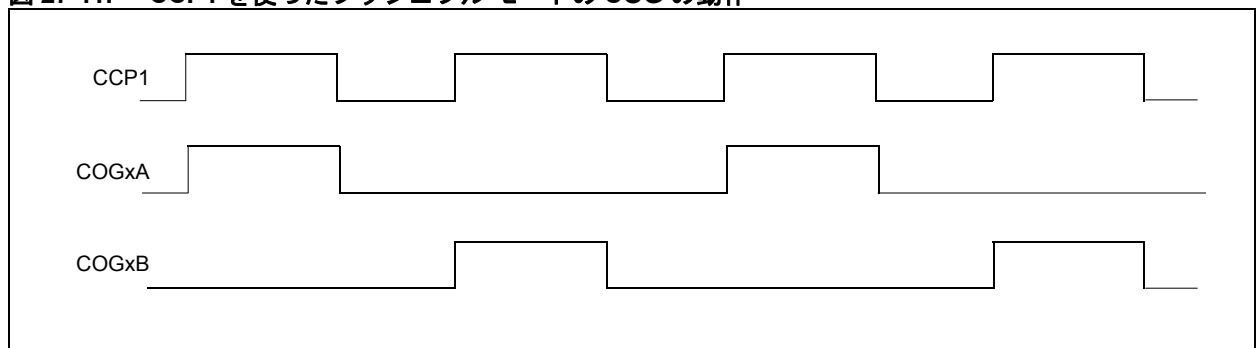


図 27-11: CCP1 を使ったプッシュプルモードの COG の動作



PIC16(L)F1764/5/8/9

図 27-12: CCP1 を使った順方向フルブリッジモードの COG の動作

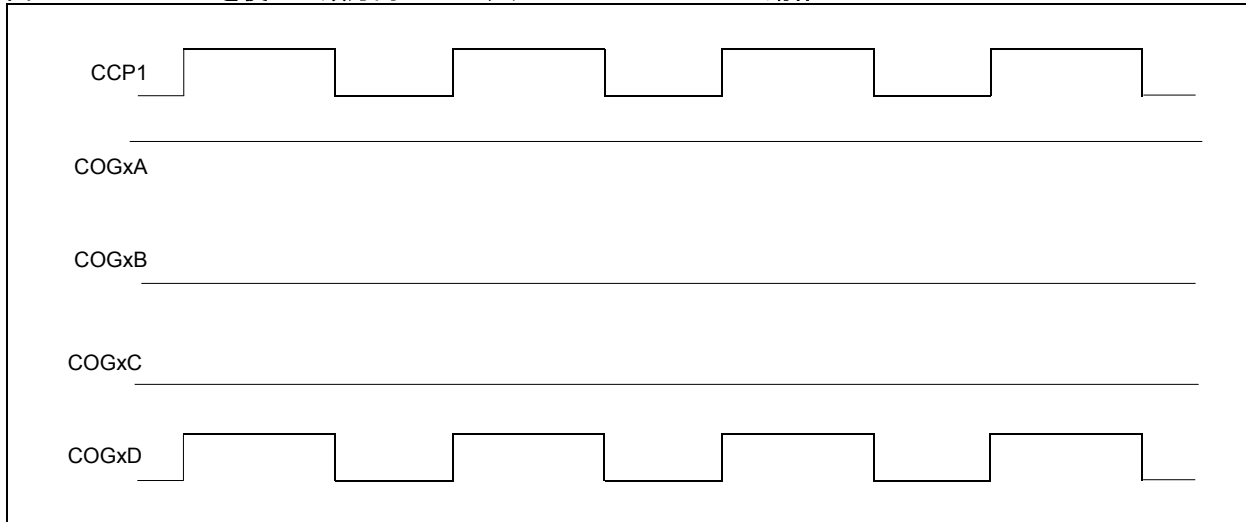
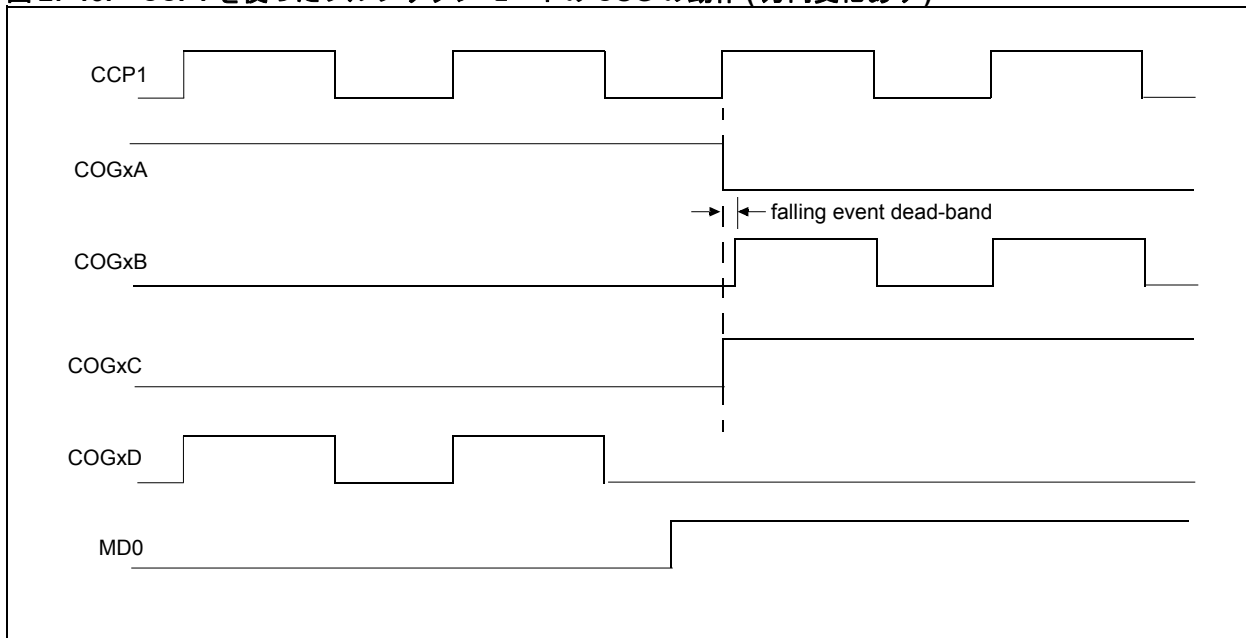


図 27-13: CCP1 を使ったフルブリッジモードの COG の動作 (方向変化あり)



27.4 クロック源

COG モジュールの各種タイマは、COG_clock を参照クロックとして使います。COG_clock クロックを使うタイマには、以下のものがあります。

- 立ち上がりおよび立ち下がリデッドバンド タイマ
- 立ち上がりおよび立ち下がリブランキング タイマ
- 立ち上がりおよび立ち下がリイベント位相遅延 タイマ

クロック源は、以下から選択できます。

- 16 MHz HFINTOSC(スリープ中もアクティブ)
- 命令クロック (Fosc/4)
- システムクロック (Fosc)

クロック源は、COGxCON0 レジスタ (レジスタ 27-1) の CS<1:0> ビットで選択します。

27.5 選択可能なイベント入力

COG は、個別に選択可能なイベント入力を自由に組み合わせることで相補波形を生成できます。イベント入力は以下の 2 種類に大別されます。

- 立ち上がりイベント入力
- 立ち下がリイベント入力

立ち上がりイベント入力は、COGxRIS0 および COGxRIS1 レジスタ (レジスタ 27-3 およびレジスタ 27-4) のビットで設定します。立ち下がリイベント入力は、COGxFIS0 および COGxF1 レジスタ (レジスタ 27-7 およびレジスタ 27-8) のビットで設定します。選択した全てのイベント入力の論理和 (OR) によって、対応するイベント信号が生成されます。図 27-7 を参照してください。

27.5.1 エッジセンスとレベルセンス

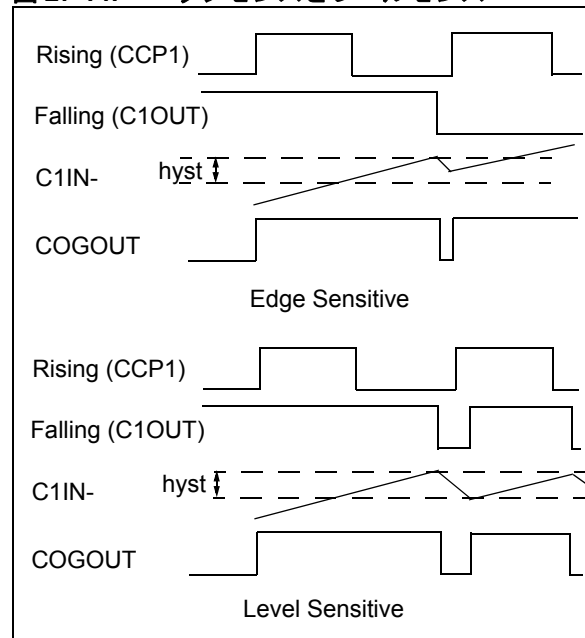
イベント入力の検出モードは、レベルセンスまたはエッジセンスのどちらかを選択できます。検出モードは各イベント入力に対して個別に選択できます。立ち上がりイベント入力の検出モードは COGxRSIM0 および COGxRSIM1 レジスタ (レジスタ 27-5 およびレジスタ 27-6) で選択します。立ち下がリイベント入力の検出モードは COGxFSIM0 および COGxFSIM1 レジスタ (レジスタ 27-9 およびレジスタ 27-10) で選択します。各ビットをセットすると、対応するイベント入力に対してエッジセンスでの検出を選択します。ビットをクリアすると、レベルセンスでの検出を選択します。

一般に、周期的な信号源で駆動されるイベントにはエッジセンスを選択し、ターゲット回路の電圧しきい値に基づくイベントにはレベルセンスを選択します。以下に 2 つの例を挙げます。

1. まず、COG 出力の周期を立ち上がりイベント入力の 50% デューティサイクルのクロックで決定し、デューティサイクルを立ち下がリイベント入力のコンパレータからの帰還電圧レベルで決定する応用回路の例を考えます。クロック入力をレベルセンスとすると、50% 未満のデューティサイクルでは正しく動作しません。レベルセンスのクロックがコンパレータからの帰還を抑制するためです。

2. 次に、デューティサイクルを 100% に近い値にして、それ以外は上の例と同じ応用回路を考えてみます。コンパレータからの帰還電圧レベルが High から Low へ遷移すると COG 駆動が OFF になりますが、ほぼ直後に周期入力によって駆動が再び ON になります。OFF サイクルが非常に短いとコンパレータ入力にヒステリシスの下限値まで到達せず、出力が変化しません。コンパレータ出力は Low のままで、High から Low への遷移がないためエッジセンスはトリガされず、COG 出力は ON に駆動された状態のままです。図 27-14 を参照してください。

図 27-14: エッジセンスとレベルセンス



PIC16(L)F1764/5/8/9

27.5.2 立ち上がりイベント

立ち上がりイベントは、PWM 出力のアクティブなデューティ サイクル期間を開始します。立ち上がりイベントとは、rising_event 出力の Low から High への遷移です。立ち上がりイベントの位相遅延とデッドバンド遅延の値が 0 の場合、主出力がただちに開始します。それ以外の場合、主出力はこれら遅延の経過後に開始します。立ち上がりイベント入力があると、以下の全てのアクションが発生します。

- 立ち上がりイベント位相遅延カウンタが開始する (有効にしている場合)
- 位相遅延の後、相補副出力をクリアする
- 立ち下がりイベント入力ブランキングを開始する (有効にしている場合)
- デッドバンド遅延を開始する (有効にしている場合)
- デッドバンド遅延の経過後、主出力をセットする

27.5.3 立ち下がりイベント

立ち下がりイベントは、PWM 出力のアクティブなデューティ サイクル期間を終了します。立ち下がりイベントとは、falling_event 出力の High から Low への遷移です。立ち下がりイベントの位相遅延とデッドバンド遅延の値が 0 の場合、相補副出力がただちに開始します。それ以外の場合、相補副出力はこれら遅延の経過後に開始します。立ち下がりイベント入力があると、以下の全てのアクションが発生します。

- 立ち下がりイベント位相遅延カウンタが開始する (有効にしている場合)
- 主出力をクリアする
- 立ち上がりイベント入力ブランキングを開始する (有効にしている場合)
- 立ち下がりイベントのデッドバンド遅延を開始する (有効にしている場合)
- デッドバンド遅延の経過後、相補副出力をセットする

27.6 出力制御

COG モジュールを無効にした場合、および有効にした直後は、COG の主出力は非アクティブで相補副出力がアクティブです。

27.6.1 出力イネーブル

COG モジュールには出力イネーブルの制御はありません。その代わりに、各デバイスピンに PPS レジスタと呼ばれる独立した出力選択制御レジスタがあります。COG の 4 つの出力は全て、PPS レジスタを設定する事で任意のピンに出力できます。

PPS レジスタで COG 出力を有効にした場合、そのピンの出力はモード (EN ビット、STR ビット、シャットダウン状態の組み合わせ) によって表 27-1 と表 27-2 のように決まります。

表 27-1: ピンの出力状態 MD<2:0> = '00x'

EN	STR ビット	シャットダウン	出力
x	0	非アクティブ	固定データをステアリング
x	1	アクティブ	シャットダウンオーバーライド
0	1	非アクティブ	非アクティブ状態
1	1	非アクティブ	アクティブな PWM 信号

表 27-2: ピンの出力状態 MD<2:0> = '001'

EN	STR ビット	シャットダウン	出力
x	x	非アクティブ	非アクティブ状態
x	x	アクティブ	シャットダウンオーバーライド
1	x	非アクティブ	アクティブな PWM 信号

27.6.2 極性制御

COG の各出力の極性は個別に選択できます。出力極性ビットをセットすると、対応する出力はアクティブ Low になります。出力極性ビットをクリアすると、対応する出力はアクティブ High になります。ただし、4 つあるシャットダウン オーバーライド モードのうち極性制御が適用されるのは 1 つのみです。詳細は [セクション 27.10「自動シャットダウン制御」](#) を参照してください。

出力極性は、COGxCON1 レジスタ ([レジスタ 27-2](#)) の POLA ~ POLD ビットで選択します。

27.7 デッドバンド制御

デッドバンド制御は、外付け電源切り換え時の貫通電流を防止するために、オーバーラップのない PWM 出力信号を生成するためのものです。デッドバンドが出力に適用されるのは、ハーフブリッジ モードの場合、およびフルブリッジ モードで方向を変更する場合のみです。

COG には 2 つのデッドバンド タイマがあります。1 つは立ち上がりイベントのデッドバンド制御に使います。もう 1 つは立ち下がりイベントのデッドバンド制御に使います。タイマモードは以下のどちらかを選択できます。

- 非同期遅延チェーン
- 同期カウンタ

デッドバンド タイマのモードは、COGxCON1 レジスタ ([レジスタ 27-2](#)) の RDBS ビット (rising_event のデッドバンド タイマ用) と FDBS ビット (falling_event のデッドバンド タイマ用) で選択します。

ハーフブリッジ モードで立ち上がりイベントが発生すると、rising_event のデッドバンドに指定した遅延時間が経過した後、選択した全ての主出力がアクティブになります。ハーフブリッジ モードでは、COGxA と COGxC が主出力です。

ハーフブリッジ モードで立ち下がりイベントが発生すると、falling_event のデッドバンドに指定した遅延時間が経過した後、選択した全ての相補副出力がアクティブになります。ハーフブリッジ モードでは、COGxB と COGxD が相補副出力です。

フルブリッジ モードでは、デッドバンドは方向を変更した場合のみ適用されます。順方向から逆方向へ変化すると、falling_event デッドバンドの経過後に変調信号が出力されます。逆方向から順方向へ変化すると、rising_event デッドバンドの経過後に変調信号が出力されます。

27.7.1 非同期遅延チェーンによるデッドバンド遅延

非同期デッドバンド遅延は、入力が直列に接続した遅延素子を伝播するのにかかる時間として設定します。1つの遅延素子が公称 5 ns です。

立ち上がりイベントの非同期デッドバンド遅延を設定するには、COGXCON0 レジスタの RDBS ビットをセットし、COGXDBR レジスタ (レジスタ 27-14) で遅延素子の数を指定してデッドバンドの長さを設定します。

立ち下がりイベントの非同期デッドバンド遅延を設定するには、COGXCON0 レジスタの FDBS ビットをセットし、COGXDBF レジスタ (レジスタ 27-15) で遅延素子の数を指定してデッドバンドの長さを設定します。

この値を 0 に設定すると、デッドバンド遅延は無効になります。

27.7.2 同期カウンタによるデッドバンド遅延

同期カウンタによるデッドバンド遅延は、COG_clock の周期を 0 からデッドバンド カウントレジスタの値までカウントする時間として設定します。デッドバンド遅延の長さは、式 27-1 で求めます。

立ち上がりイベントの同期デッドバンド遅延を設定するには、COGXCON0 レジスタの RDBS ビットをクリアし、COGXDBR カウントレジスタで COG_clock 周期の数を指定してデッドバンドの長さを設定します。

立ち下がりイベントの同期デッドバンド遅延を設定するには、COGXCON0 レジスタの FDBS ビットをクリアし、COGXDBF カウントレジスタで COG_clock 周期の数を指定してデッドバンドの長さを設定します。

この値を 0 に設定すると、デッドバンド遅延は無効になります。

27.7.3 同期カウンタによるデッドバンド遅延の不定

非同期入力による立ち上がりおよび立ち下がりイベントでデッドバンド カウンタをトリガする場合、同期カウンタのデッドバンド遅延は不定です。この不定値は、最大で COG_clock の 1 周期に等しくなります。詳細は例 27-1 を参照してください。

イベント入力が位相遅延のない非同期入力の場合、デッドバンド遅延が不定となるのを避けるため非同期遅延チェーンによるデッドバンドを使います。

27.7.4 立ち上がりイベントのデッドバンド

立ち上がりイベントのデッドバンドは、相補副出力が OFF になってから主出力が ON になるまでの遅延です。立ち上がりイベントのデッドバンド遅延は、rising_event 出力が真になると開始します。

立ち下がりエッジのデッドバンド遅延の設定方法は、セクション 27.7.1「非同期遅延チェーンによるデッドバンド遅延」とセクション 27.7.2「同期カウンタによるデッドバンド遅延」を参照してください。

27.7.5 立ち下がりイベントのデッドバンド

立ち下がりイベントのデッドバンドは、主出力が OFF になってから相補副出力が ON になるまでの遅延です。立ち下がりイベントのデッドバンド遅延は、falling_event 出力が真になると開始します。

立ち下がりエッジのデッドバンド遅延の設定方法は、セクション 27.7.1「非同期遅延チェーンによるデッドバンド遅延」とセクション 27.7.2「同期カウンタによるデッドバンド遅延」を参照してください。

27.7.6 デッドバンドの重なり

デッドバンドが重なるのは、以下の 2 つの可能性あります。

- 立ち上がりから立ち下がり
- 立ち下がりから立ち上がり

27.7.6.1 立ち上がりから立ち下がりの重なり

これは、立ち上がりイベント デッドバンド カウンタがカウント中に立ち下がりイベントが発生した場合です。この場合、主出力が抑止され、立ち下がりイベント デッドバンドの長さ分デッドバンド遅延が延長されます。延長されたデッドバンド遅延が終了すると、相補副出力が真になります。

27.7.6.2 立ち下がりから立ち上がりの重なり

これは、立ち下がりイベント デッドバンド カウンタがカウント中に立ち上がりイベントが発生した場合です。この場合、相補副出力が抑止され、立ち上がりイベント デッドバンドの長さ分デッドバンド遅延が延長されます。延長されたデッドバンド遅延が終了すると、主出力が真になります。

27.8 ブランキング制御

入力ブランキングは、短時間イベント入力をマスク (ブランキング) する機能です。これにより、パワーデバイスの ON/OFF に起因する過渡サージによって入力イベントが誤って生成されるのを防ぎます。

COG には 2 つのブランキング カウンタがあり、1 つは立ち上がりイベントでトリガされ、もう 1 つは立ち下がりイベントでトリガされます。これらのカウンタは、トリガするイベントとマスクされるイベントが互い違いです。立ち下がりイベント ブランキング カウンタは立ち上がり入力イベントをマスクするのに使い、立ち上がりイベント ブランキング カウンタは立ち下がり入力イベントをマスクするのに使います。ブランキングが開始すると、対応するブランキング カウンタで指定した長さ分ブランキングが持続します。

ブランキング遅延は、COG_clock の周期を 0 からブランキング カウントレジスタの値までカウントする時間として設定します。ブランキング遅延の長さは、式 27-1 で求めます。

27.8.1 立ち下がリイベント ブランキングによる立ち上がりイベントのマスク

立ち下がリイベント ブランキング カウンタのカウント中は、立ち上がりイベント入力があっても立ち上がりイベントはトリガされません。立ち下がリイベント ブランキングは、立ち上がりイベント出力が偽になると開始します。

立ち下がリイベント ブランキングの長さは、COGxBLKF レジスタ (レジスタ 27-17) の値で設定します。ブランキングの長さは、式 27-1 で求めます。

COGxBLKF の値が 0 の場合は立ち下がリイベント ブランキングが無効になり、ブランキング カウンタの出力は真となるため、イベント信号がそのままイベントトリガ回路へ通過します。

27.8.2 立ち上がりイベント ブランキングによる立ち下がリイベントのマスク

立ち上がりイベント ブランキング カウンタのカウント中は、立ち下がリイベント入力があっても立ち下がリイベントはトリガされません。立ち上がりイベント ブランキングは、立ち下がリイベント出力が偽になると開始します。

立ち上がりイベント ブランキングの長さは、COGxBLKR レジスタ (レジスタ 27-16) の値で設定します。

COGxBLKR の値が 0 の場合は立ち上がりイベント ブランキングが無効になり、ブランキング カウンタの出力は真となるため、イベント信号がそのままイベントトリガ回路へ通過します。

27.8.3 ブランキング遅延の不定

COG_clock に対して非同期的な立ち上がりおよび立ち下がリイベント入力でのブランキング カウンタをトリガする場合、ブランキングの長さは不定です。この不定値は、最大で COG_clock の 1 周期に等しくなります。詳細は、式 27-1 と例 27-1 を参照してください。

27.9 位相遅延

立ち上がりイベントと立ち下がリイベントの片方または両方のアサートを遅らせる事ができます。この機能を使うには、COGxPHR または COGxPHF 位相遅延カウンタレジスタ (レジスタ 27-18、レジスタ 27-19 参照) に 0 以外の値を設定します。CCP1 を用いた COG 動作で位相遅延を有効にした場合のタイミングは、図 27-10 を参照してください。立ち上がりイベント入力の信号が遷移してからイベントが実際にアサートされるまでの遅延は、デッドバンドおよびブランキング遅延と同じ方法で求めます。式 27-1 を参照してください。

位相遅延カウンタ値が 0 の場合は位相遅延が無効になり、位相遅延カウンタの出力は真となるため、イベント信号がそのまま相補副出力のドライバフロップへ通過します。

27.9.1 不定値の累積

複数の段が連続しても、遅延の不定値が 1 COG_clock を超える事はありません。例えばブランキング段の後に位相遅延段があり、ブランキングまたは位相遅延段の後にデッドバンド段があり、デッドバンド段の後にブランキング段がある場合を考えてみます。ある段を有効にすると、その段の出力は必ず COG_clock と同期するため、以降の段で遅延が不定となる可能性はありません。

式 27-1: 位相遅延、デッドバンド遅延、ブランキング遅延の計算

$$T_{\min} = \frac{\text{Count}}{F_{\text{COG_clock}}}$$

$$T_{\max} = \frac{\text{Count} + 1}{F_{\text{COG_clock}}}$$

$$T_{\text{uncertainty}} = T_{\max} - T_{\min}$$

また:

$$T_{\text{uncertainty}} = \frac{1}{F_{\text{COG_clock}}}$$

T	Count
立ち上がり位相遅延	COGxPHR
立ち下がリ位相遅延	COGxPHF
立ち上がりデッドバンド	COGxDBR
立ち下がリデッドバンド	COGxDBF
立ち上がりイベントブランキング	COGxBLKR
立ち下がリイベントブランキング	COGxBLKF

例 27-1: タイマの不定値

条件:

$$Count = Ah = 10d$$

$$F_{COG_Clock} = 8MHz$$

従って:

$$T_{uncertainty} = \frac{1}{F_{COG_clock}}$$

$$= \frac{1}{8MHz} = 125ns$$

証明:

$$T_{min} = \frac{Count}{F_{COG_clock}}$$

$$= 125ns \cdot 10d = 1.25\mu s$$

$$T_{max} = \frac{Count + 1}{F_{COG_clock}}$$

$$= 125ns \cdot (10d + 1)$$

$$= 1.375\mu s$$

従って:

$$T_{uncertainty} = T_{max} - T_{min}$$

$$= 1.375\mu s - 1.25\mu s$$

$$= 125ns$$

自動再起動が無効の場合、ASE ビットをソフトウェアでクリアした後の最初の立ち上がりイベントまでシャットダウン状態が持続します。

自動再起動が有効の場合、ASE ビットが自動的にクリアされ、シャットダウン入力クリアされた後の最初の立ち上がりイベントから動作が再開します。[図 27-15](#) と [セクション 27.10.3.2 「自動再起動」](#) を参照してください。

27.10.1.2 外部入力によるシャットダウン

外部入力をシャットダウン要因に使うと、フォルト条件時に COG の動作を最も短時間で安全に停止できます。選択したシャットダウン入力のいずれかが真になると出力駆動ラッチがリセットされ、COG 出力はソフトウェアによる遅延なしに選択したオーバーライドレベルにただちに遷移します。

シャットダウン条件は、任意の入力の組み合わせによって生成できます。選択した入力が Low になるとシャットダウンが発生します。シャットダウン要因には、以下のものを指定できます。

- COGxINPPS制御レジスタで選択した任意の入力ピン
- コンパレータ 1
- コンパレータ 2
- コンパレータ 3
- コンパレータ 4
- CLC2 の出力
- Timer2 の出力
- Timer4 の出力

シャットダウン要因は、COGxASD1 レジスタ ([レジスタ 27-12](#)) のビットで個別に選択できます。

Note: シャットダウン入力はエッジセンスではなく、レベルセンスです。シャットダウン入力レベルが持続している間は、自動シャットダウンを無効にしない限り、シャットダウン状態をクリアする事はできません。

27.10 自動シャットダウン制御

自動シャットダウンとは、COG 出力レベルを特定の信号でただちにオーバーライドして、回路を安全にシャットダウンする手法です。

シャットダウン状態は、自動的にクリアする事も、ソフトウェアでクリアするまでその状態を保持する事もできます。どちらの場合も、シャットダウンをクリアした後の最初の立ち上がりイベントまでシャットダウンオーバーライドは有効のままです。

27.10.1 シャットダウン

シャットダウン状態への移行方法には、以下の 2 つがあります。

- ソフトウェアによるシャットダウン
- 外部入力によるシャットダウン

27.10.1.1 ソフトウェアによるシャットダウン

COGxASD0 レジスタ ([レジスタ 27-11](#)) の ASE ビットをセットすると、COG がシャットダウン状態に移行します。

27.10.2 ピン オーバーライド レベル

シャットダウンが有効中に出力ピンに駆動されるレベルは、COGxASD0 レジスタ ([レジスタ 27-11](#)) の ASDAC<1:0>および ASDBC<1:0> ビットで制御します。COGxA と COGxC のオーバーライドレベルは ASDAC<1:0> で制御し、COGxB と COGxD のオーバーライドレベルは ASDBC<1:0> で制御します。各出力ペアに対して、以下の 4 種類のオーバーライドが可能です。

- Low に強制
- High に強制
- 3 ステート
- PWM 非アクティブ状態 (立ち下がりがイベント発生時と同じ状態)

Note: 極性制御は、オーバーライドレベルを Low または High に強制した場合は適用されませんが、PWM 非アクティブ状態の場合は適用されます。

PIC16(L)F1764/5/8/9

27.10.3 自動シャットダウン後の再起動

自動シャットダウン イベントが発生した場合、動作を再開するには以下の2つの方法があります。

- ソフトウェア制御による再起動
- 自動再起動

再起動の方法は COGxASD0 レジスタの ARSEN ビットで選択します。図 27-15 に、ソフトウェア制御による再起動と自動再起動の波形を示します。

27.10.3.1 ソフトウェア制御による再起動

COGxASD0 レジスタの ARSEN ビットをクリアした場合、自動シャットダウン イベント後に COG 動作を再開するには ASE ビットをソフトウェアでクリアする必要があります。

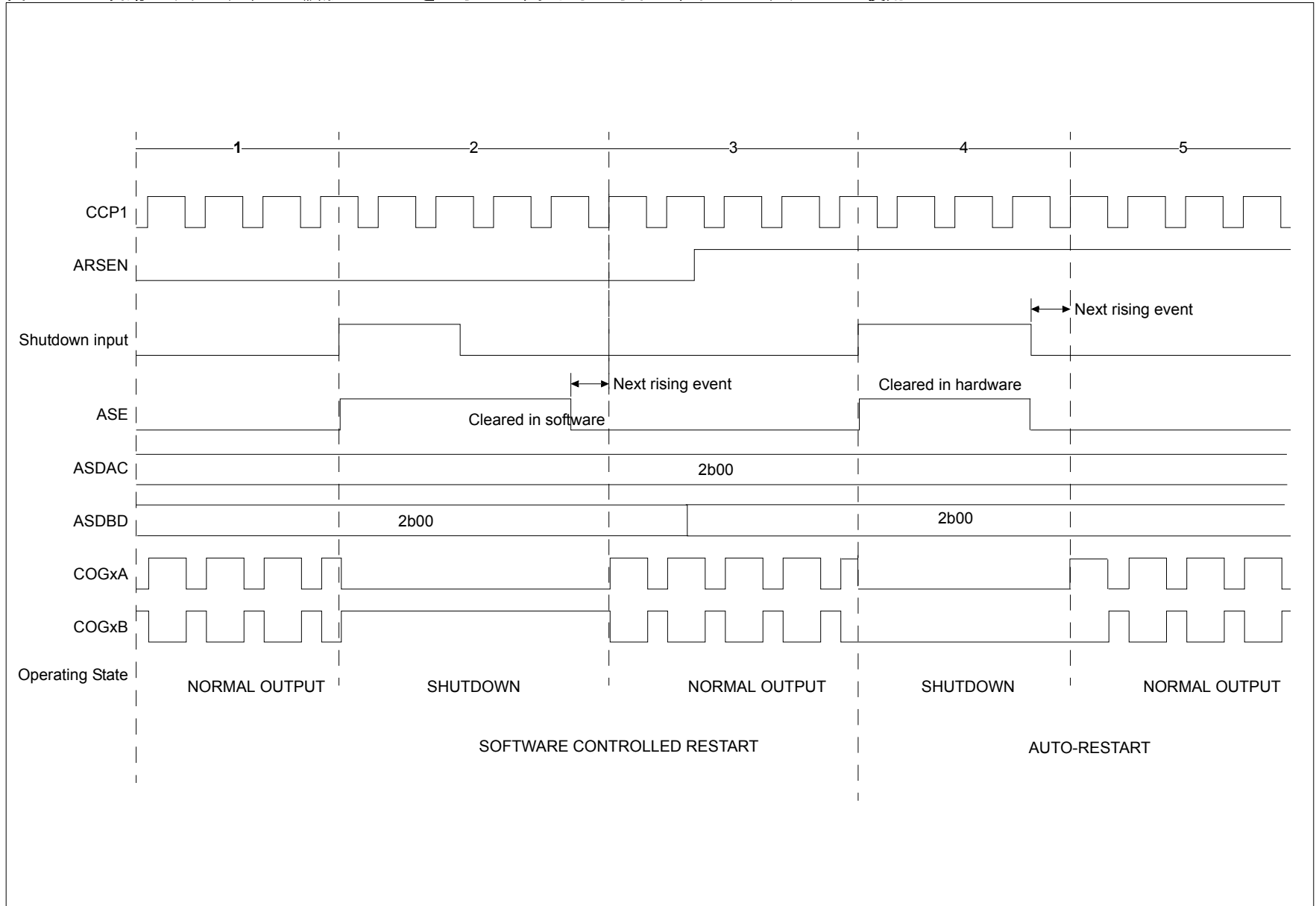
ASE ビットのクリア後、最初の立ち上がりイベントで COG が動作を再開します。シャットダウン状態を解除するには、選択したシャットダウン入力が全て偽である必要があります。それ以外の場合、ASE ビットはセットされたままです。

27.10.3.2 自動再起動

COGxASD0 レジスタの ARSEN ビットをセットしている場合、COG は自動シャットダウン状態から自動的に再起動します。

選択したシャットダウン入力が全て偽になった後、最初の立ち上がりイベントで ASE ビットが自動的にクリアされ、COG が動作を再開します。

図 27-15: 自動シャットダウンの波形 – CCP1 を立ち上がりおよび立ち下がリイベント入力として使用



27.11 バッファの更新

COGの動作中、位相遅延、デッドバンド、ブランキングの各カウントレジスタへの書き込みタイミングに遅延があると予期しない動作となる事があるため、これらレジスタの変更は同時に実行する必要があります。そこで、COGxCON0 レジスタのLD ビットを使って位相遅延、デッドバンド、ブランキング カウントレジスタの内容を同時にバッファに書き込むようにします。

COG モジュールを有効にする前なら、カウントレジスタへ書き込んだ内容がただちにカウントバッファに書き込まれるため、LD ビットを使う必要はありません。しかし COG が有効の場合、カウントレジスタへ書き込んで LD ビットをセットするまでカウントバッファは更新されません。LD ビットをセットすると、位相遅延、デッドバンド、ブランキング カウントレジスタの値が COG 動作に同期してそれぞれのバッファに転送されます。転送が完了すると、LD ビットはハードウェアでクリアされます。

27.12 入力および出力ピンの選択

COGの入力には、1本のデバイスピンを選択できます。この入力、立ち上がりおよび立ち下がりイベント入力またはフォルト入力として使えます。このピンは、COGxINPPS レジスタで選択します。xxxPPS レジスタ ([レジスタ 12-1](#)) と RxyPPS レジスタ ([レジスタ 12-2](#)) を参照してください。

COG 出力を有効にするには、そのピンの PPS 制御レジスタを使います。同じ出力を複数のピンに割り当て等、ピンへの出力は任意の組み合わせが可能です。詳細は、RxyPPS 制御レジスタおよび [セクション 12.2 「PPS 出力」](#) を参照してください。

27.13 スリープ中の動作

COG_clock、立ち上がりイベント、立ち下がりイベントの各入力が入力アクティブであれば、COG はスリープ中も動作を継続します。

COG を有効にして HFINTOSC を COG_clock のクロック源に選択している場合、HFINTOSC はスリープ中もアクティブなままです。

27.14 COG の設定

COG を立ち上がりイベント入力に確実に同期して起動させるには、以下の手順で COG を設定します。

1. COG フォルトまたはイベント入力にピンを使う場合、COGxINPPS レジスタでピンを選択する。
2. COG 機能に使うピンに関連する全ての ANSEL レジスタビットをクリアする。
3. COG 出力に対応する TRIS 制御ビットをセットして、全て入力に設定する (後で COG モジュールが必要に応じて出力ドライバを有効にします)。
4. EN ビットがまだクリアされていない場合、クリアする。

5. COGxDBR および COGxDBF レジスタで目的のデッドバンド遅延の長さを設定し、COGxCON1 レジスタの RDBS および FDBS ビットでタイムモードを選択する。
6. COGxBLKR および COGxBLKF レジスタでブランキング遅延の長さを設定する。
7. COGxPHR および COGxPHF レジスタで位相遅延の長さを設定する。
8. COGxASD1 レジスタでシャットダウン要因を選択する。
9. COGxASD0 自動シャットダウン レジスタを以下の通り設定する。
 - 両方の出力オーバーライド レベルを選択する (通常の起動もシャットダウン状態から開始するため、この設定は自動シャットダウンを使わない場合にも必要です)
 - ASE ビットをセットし、ARSEN ビットをクリアする
10. COGxRIS0、COGxRIS1、COGxFIS0、COGxFIS1 の各レジスタで立ち上がりおよび立ち下がりイベント入力を選択する。
11. COGxRSIM0、COGxRSIM1、COGxFSIM0、COGxFSIM1 の各レジスタで立ち上がりおよび立ち下がりイベントの検出モードを選択する。
12. COGxCON1 レジスタで以下の内容を設定する。
 - 各出力の極性を設定する
 - デッドバンドのタイミング源を選択する
13. COGxCON0 レジスタで以下の内容を設定する。
 - 動作モードを設定する
 - クロック源を選択する
14. ステアリングモードの1つを選択する場合、COGxSTR レジスタで以下の内容を設定する。
 - 使う出力のステアリング ビットをセットする
 - 固定レベルを設定する
15. EN ビットをセットする。
16. COG出力に使うピンをPPSレジスタで設定する。
17. 自動再起動を使う場合、ARSEN ビットをセットすると ASE ビットが自動的にクリアされる。自動再起動を使わない場合、ASE ビットをクリアして COG を起動する。

27.15 レジスタ定義 :COG 制御

表 27-3 に COG 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1 「レジスタおよびビット命名法」](#) を参照してください。

表 27-3:

周辺モジュール	ビット名の接頭辞
COG1	G1
COG2 ⁽¹⁾	G2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 27-1: COGxCON0: COG 制御レジスタ 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD	—	CS<1:0>		MD<2:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7 **EN:** COGx イネーブルビット

- 1 = モジュールを有効にする
- 0 = モジュールを無効にする

bit 6 **LD:** COGx バッファ書き込みビット

- 1 = 次の入カイベントで位相遅延、ブランキング、デッドバンドバッファにレジスタ値を書き込む
- 0 = バッファへのレジスタ値の転送が完了している

bit 5 **未実装:** 「0」として読み出し

bit 4-3 **CS<1:0>:** COGx クロック選択ビット

- 11 = 予約済み使用不可
- 10 = COG_clock に HFINTOSC を使う (スリープ中も有効)
- 01 = COG_clock に Fosc を使う
- 00 = COG_clock に Fosc/4 を使う

bit 2-0 **MD<2:0>:** COGx モード選択ビット

- 11x = 予約済み、使用不可
- 101 = COG 出力の動作をプッシュプルモードにする
- 100 = COG 出力の動作をハーフブリッジモードにする
- 011 = COG 出力の動作を逆方向フルブリッジモードにする
- 010 = COG 出力の動作を順方向フルブリッジモードにする
- 001 = COG 出力の動作を同期 PWM ステアリングモードにする
- 000 = COG 出力の動作を PWM ステアリングモードにする

PIC16(L)F1764/5/8/9

レジスタ 27-2: COGxCON1: COG 制御レジスタ 1

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RDBS	FDBS	—	—	POLD	POLC	POLB	POLA
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

- bit 7 **RDBS:** COGx 立ち上がりイベント デッドバンド タイミング源選択ビット
1 = 遅延チェーンと COGxDBR を使ってデッドバンド遅延を生成する
0 = COGx_clock と COGxDBR を使ってデッドバンド遅延を生成する
- bit 6 **FDBS:** COGx 立ち下がりイベント デッドバンド タイミング源選択ビット
1 = 遅延チェーンと COGxDBF を使ってデッドバンド遅延を生成する
0 = COGx_clock と COGxDBF を使ってデッドバンド遅延を生成する
- bit 5-4 **未実装:** 「0」として読み出し
- bit 3 **POLD:** COGxD 出力極性制御ビット
1 = COGxD 出力をアクティブ Low にする
0 = COGxD 出力をアクティブ Low にする
- bit 2 **POLC:** COGxC 出力極性制御ビット
1 = COGxC 出力をアクティブ Low にする
0 = COGxC 出力をアクティブ High にする
- bit 1 **POLB:** COGxB 出力極性制御ビット
1 = COGxB 出力をアクティブ Low にする
0 = COGxB 出力をアクティブ High にする
- bit 0 **POLA:** COGxA 出力極性制御ビット
1 = COGxA 出力をアクティブ Low にする
0 = COGxA 出力をアクティブ High にする

レジスタ 27-3: COGxRIS0: COG 立ち上がりイベント入力選択レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7	RIS7: COGx 立ち上がりイベント入力源 7 イネーブルビット 1 = PWM3 出力を立ち上がりイベント入力として有効にする 0 = PWM3 出力は立ち上がりイベントに影響を与えない
bit 6	RIS6: COGx 立ち上がりイベント入力源 6 イネーブルビット 1 = CCP2 出力を立ち上がりイベント入力として有効にする 0 = CCP2 出力は立ち上がりイベントに影響を与えない
bit 5	RIS5: COGx 立ち上がりイベント入力源 5 イネーブルビット 1 = CCP1 出力を立ち上がりイベント入力として有効にする 0 = CCP1 出力は立ち上がりイベントに影響を与えない
bit 4	RIS4: COGx 立ち上がりイベント入力源 4 イネーブルビット 1 = コンパレータ 4 出力を立ち上がりイベント入力として有効にする 0 = コンパレータ 4 出力は立ち上がりイベントに影響を与えない
bit 3	RIS3: COGx 立ち上がりイベント入力源 3 イネーブルビット 1 = コンパレータ 3 出力を立ち上がりイベント入力として有効にする 0 = コンパレータ 3 出力は立ち上がりイベントに影響を与えない
bit 2	RIS2: COGx 立ち上がりイベント入力源 2 イネーブルビット 1 = コンパレータ 2 出力を立ち上がりイベント入力として有効にする 0 = コンパレータ 2 出力は立ち上がりイベントに影響を与えない
bit 1	RIS1: COGx 立ち上がりイベント入力源 1 イネーブルビット 1 = コンパレータ 1 出力を立ち上がりイベント入力として有効にする 0 = コンパレータ 1 出力は立ち上がりイベントに影響を与えない
bit 0	RIS0: COGx 立ち上がりイベント入力源 0 イネーブルビット 1 = COGxINPPS 制御レジスタで選択したピンを立ち上がりイベント入力として有効にする 0 = COGxINPPS 制御レジスタで選択したピンは立ち上がりイベントに影響を与えない

PIC16(L)F1764/5/8/9

レジスタ 27-4: COGxRIS1: COG 立ち上がりイベント入力選択レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

- bit 7-6 **未実装:** 「0」として読み出し
- bit 5 **RIS13:** COGx 立ち上がりイベント入力源 13 イネーブルビット
1 = CLC3 出力を立ち上がりイベント入力として有効にする
0 = CLC3 出力は立ち上がりイベントに影響を与えない
- bit 4 **RIS12:** COGx 立ち上がりイベント入力源 12 イネーブルビット
1 = CLC2 出力を立ち上がりイベント入力として有効にする
0 = CLC2 出力は立ち上がりイベントに影響を与えない
- bit 3 **RIS11:** COGx 立ち上がりイベント入力源 11 イネーブルビット
1 = CLC1 出力を立ち上がりイベント入力として有効にする
0 = CLC1 出力は立ち上がりイベントに影響を与えない
- bit 2 **RIS10:** COGx 立ち上がりイベント入力源 10 イネーブルビット
1 = PWM6 出力を立ち上がりイベント入力として有効にする
0 = PWM6 出力は立ち上がりイベントに影響を与えない
- bit 1 **RIS9:** COGx 立ち上がりイベント入力源 9 イネーブルビット
1 = PWM5 出力を立ち上がりイベント入力として有効にする
0 = PWM5 出力は立ち上がりイベントに影響を与えない
- bit 0 **RIS8:** COGx 立ち上がりイベント入力源 8 イネーブルビット
1 = PWM4 出力を立ち上がりイベント入力として有効にする
0 = PWM4 出力は立ち上がりイベントに影響を与えない

レジスタ 27-5: COGxRSIM0: COG 立ち上がりイベント入力モードレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7 **RSIM7:** COGx 立ち上がりイベント入力 7 モードビット

RIS7 = 1:

1 = PWM3 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = PWM3 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS7 = 0:

PWM3 出力は立ち上がりイベントに影響を与えない

bit 6 **RSIM6:** COGx 立ち上がりイベント入力 6 モードビット

RIS6 = 1:

1 = CCP2 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = CCP2 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS6 = 0:

CCP2 出力は立ち上がりイベントに影響を与えない

bit 5 **RSIM5:** COGx 立ち上がりイベント入力 5 モードビット

RIS5 = 1:

1 = CCP1 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = CCP1 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS5 = 0:

CCP1 出力は立ち上がりイベントに影響を与えない

bit 4 **RSIM4:** COGx 立ち上がりイベント入力 4 モードビット

RIS4 = 1:

1 = コンパレータ 4 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = コンパレータ 4 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS4 = 0:

コンパレータ 4 出力は立ち上がりイベントに影響を与えない

bit 3 **RSIM3:** COGx 立ち上がりイベント入力 3 モードビット

RIS3 = 1:

1 = コンパレータ 3 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = コンパレータ 3 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS3 = 0:

コンパレータ 3 出力は立ち上がりイベントに影響を与えない

bit 2 **RSIM2:** COGx 立ち上がりイベント入力 2 モードビット

RIS2 = 1:

1 = コンパレータ 2 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = コンパレータ 2 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS2 = 0:

コンパレータ 2 出力は立ち上がりイベントに影響を与えない

bit 1 **RSIM1:** COGx 立ち上がりイベント入力 1 モードビット

RIS1 = 1:

1 = コンパレータ 1 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = コンパレータ 1 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS1 = 0:

コンパレータ 1 出力は立ち上がりイベントに影響を与えない

bit 0 **RSIM0:** COGx 立ち上がりイベント入力 0 モードビット

RIS0 = 1:

1 = COGxINPPS レジスタで選択したピンが Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = COGxINPPS レジスタで選択したピンの High レベルでただちに立ち上がりイベントを発生させる

RIS0 = 0:

COGxINPPS レジスタで選択したピンは立ち上がりイベントに影響を与えない

PIC16(L)F1764/5/8/9

レジスタ 27-6: COGxRSIM1: COG 立ち上がりイベント入力モードレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7-6 **未実装**: 「0」として読み出し

bit 5 **RSIM13**: COGx 立ち上がりイベント入力 13 モードビット

RIS13 = 1:

1 = CLC3 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = CLC3 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS13 = 0:

CLC3 出力は立ち上がりイベントに影響を与えない

bit 4 **RSIM12**: COGx 立ち上がりイベント入力 12 モードビット

RIS12 = 1:

1 = CLC2 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = CLC2 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS12 = 0:

CLC2 出力は立ち上がりイベントに影響を与えない

bit 3 **RSIM11**: COGx 立ち上がりイベント入力 11 モードビット

RIS11 = 1:

1 = CLC1 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = CLC1 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS11 = 0:

CLC1 出力は立ち上がりイベントに影響を与えない

bit 2 **RSIM10**: COGx 立ち上がりイベント入力 10 モードビット

RIS10 = 1:

1 = PWM6 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = PWM6 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS10 = 0:

PWM6 出力は立ち上がりイベントに影響を与えない

bit 1 **RSIM9**: COGx 立ち上がりイベント入力 9 モードビット

RIS9 = 1:

1 = PWM5 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = PWM5 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS9 = 0:

PWM5 出力は立ち上がりイベントに影響を与えない

bit 0 **RSIM8**: COGx 立ち上がりイベント入力 8 モードビット

RIS8 = 1:

1 = PWM4 出力が Low から High へ遷移すると、立ち上がりイベント位相遅延の後、立ち上がりイベントを発生させる

0 = PWM4 出力の High レベルでただちに立ち上がりイベントを発生させる

RIS8 = 0:

PWM4 出力は立ち上がりイベントに影響を与えない

レジスタ 27-7: COGxFIS0: COG 立ち下がリイベント入力選択レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

- bit 7 **FIS7:** COGx 立ち下がリイベント入力 7 イネーブルビット
 1 = PWM3 出力を立ち下がリイベント入力として有効にする
 0 = PWM3 出力は立ち下がリイベントに影響を与えない
- bit 6 **FIS6:** COGx 立ち下がリイベント入力 6 イネーブルビット
 1 = CCP2 出力を立ち下がリイベント入力として有効にする
 0 = CCP2 出力は立ち下がリイベントに影響を与えない
- bit 5 **FIS5:** COGx 立ち下がリイベント入力 5 イネーブルビット
 1 = CCP1 出力を立ち下がリイベント入力として有効にする
 0 = CCP1 出力は立ち下がリイベントに影響を与えない
- bit 4 **FIS4:** COGx 立ち下がリイベント入力 4 イネーブルビット
 1 = コンパレータ 4 出力を立ち下がリイベント入力として有効にする
 0 = コンパレータ 4 出力は立ち下がリイベントに影響を与えない
- bit 3 **FIS3:** COGx 立ち下がリイベント入力 3 イネーブルビット
 1 = コンパレータ 3 出力を立ち下がリイベント入力として有効にする
 0 = コンパレータ 3 出力は立ち下がリイベントに影響を与えない
- bit 2 **FIS2:** COGx 立ち下がリイベント入力 2 イネーブルビット
 1 = コンパレータ 2 出力を立ち下がリイベント入力として有効にする
 0 = コンパレータ 2 出力は立ち下がリイベントに影響を与えない
- bit 1 **FIS1:** COGx 立ち下がリイベント入力 1 イネーブルビット
 1 = コンパレータ 1 出力を立ち下がリイベント入力として有効にする
 0 = コンパレータ 1 出力は立ち下がリイベントに影響を与えない
- bit 0 **FIS0:** COGx 立ち下がリイベント入力 0 イネーブルビット
 1 = COGxINPPS 制御レジスタで選択したピンを立ち下がリイベント入力として有効にする
 0 = COGxINPPS 制御レジスタで選択したピンは立ち下がリイベントに影響を与えない

PIC16(L)F1764/5/8/9

レジスタ 27-8: COGxFIS1: COG 立ち下がリイベント入力選択レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

- bit 7-6 **未実装:** 「0」として読み出し
- bit 5 **FIS13:** COGx 立ち下がリイベント入力 13 イネーブルビット
1 = CLC3 出力を立ち下がリイベント入力として有効にする
0 = CLC3 出力は立ち下がリイベントに影響を与えない
- bit 4 **FIS12:** COGx 立ち下がリイベント入力 12 イネーブルビット
1 = CLC2 出力を立ち下がリイベント入力として有効にする
0 = CLC2 出力は立ち下がリイベントに影響を与えない
- bit 3 **FIS11:** COGx 立ち下がリイベント入力 11 イネーブルビット
1 = CLC1 出力を立ち下がリイベント入力として有効にする
0 = CLC1 出力は立ち下がリイベントに影響を与えない
- bit 2 **FIS10:** COGx 立ち下がリイベント入力 10 イネーブルビット
1 = PWM6 出力を立ち下がリイベント入力として有効にする
0 = PWM6 出力は立ち下がリイベントに影響を与えない
- bit 1 **FIS9:** COGx 立ち下がリイベント入力 9 イネーブルビット
1 = PWM5 出力を立ち下がリイベント入力として有効にする
0 = PWM5 出力は立ち下がリイベントに影響を与えない
- bit 0 **FIS8:** COGx 立ち下がリイベント入力 8 イネーブルビット
1 = PWM4 出力を立ち下がリイベント入力として有効にする
0 = PWM4 出力は立ち下がリイベントに影響を与えない

レジスタ 27-9: COGxFSIM0: COG 立ち下がリイベント入力モードレジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7 **FSIM7:** COGx 立ち下がリイベント入力 7 モードビット

FSIM7 = 1:

1 = PWM3 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = PWM3 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM7 = 0:

PWM3 出力は立ち下がリイベントに影響を与えない

bit 6 **FSIM6:** COGx 立ち下がリイベント入力 6 モードビット

FSIM6 = 1:

1 = CCP2 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = CCP2 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM6 = 0:

CCP2 出力は立ち下がリイベントに影響を与えない

bit 5 **FSIM5:** COGx 立ち下がリイベント入力 5 モードビット

FSIM5 = 1:

1 = CCP1 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = CCP1 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM5 = 0:

CCP1 出力は立ち下がリイベントに影響を与えない

bit 4 **FSIM4:** COGx 立ち下がリイベント入力 4 モードビット

FSIM4 = 1:

1 = コンパレータ 4 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = コンパレータ 4 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM4 = 0:

コンパレータ 4 出力は立ち下がリイベントに影響を与えない

bit 3 **FSIM3:** COGx 立ち下がリイベント入力 3 モードビット

FSIM3 = 1:

1 = コンパレータ 3 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = コンパレータ 3 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM3 = 0:

コンパレータ 3 出力は立ち下がリイベントに影響を与えない

bit 2 **FSIM2:** COGx 立ち下がリイベント入力 2 モードビット

FSIM2 = 1:

1 = コンパレータ 2 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = コンパレータ 2 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM2 = 0:

コンパレータ 2 出力は立ち下がリイベントに影響を与えない

bit 1 **FSIM1:** COGx 立ち下がリイベント入力 1 モードビット

FSIM1 = 1:

1 = コンパレータ 1 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = コンパレータ 1 出力の Low レベルでただちに立ち下がリイベントを発生させる

FSIM1 = 0:

コンパレータ 1 出力は立ち下がリイベントに影響を与えない

bit 0 **FSIM0:** COGx 立ち下がリイベント入力 0 モードビット

FSIM0 = 1:

1 = COGxINPPS 制御レジスタで選択したピンが High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる
0 = COGxINPPS 制御レジスタで選択したピンの Low レベルでただちに立ち下がリイベントを発生させる

FSIM0 = 0:

COGxINPPS 制御レジスタで選択したピンは立ち下がリイベントに影響を与えない

PIC16(L)F1764/5/8/9

レジスタ 27-10: COGxFSIM1: COG 立ち下がリイベント入力モードレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7-6	未実装: 「0」として読み出し
bit 5	FSIM13: COGx 立ち下がリイベント入力 13 モードビット FIS13 = 1: 1 = CLC3 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = CLC3 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS13 = 0: CLC3 出力は立ち下がリイベントに影響を与えない
bit 4	FSIM12: COGx 立ち下がリイベント入力 12 モードビット FIS12 = 1: 1 = CLC2 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = CLC2 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS12 = 0: CLC2 出力は立ち下がリイベントに影響を与えない
bit 3	FSIM11: COGx 立ち下がリイベント入力 11 モードビット FIS11 = 1: 1 = CLC1 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = CLC1 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS11 = 0: CLC1 出力は立ち下がリイベントに影響を与えない
bit 2	FSIM10: COGx 立ち下がリイベント入力 10 モードビット FIS10 = 1: 1 = PWM6 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = PWM6 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS10 = 0: コンパレータ 2 出力は立ち下がリイベントに影響を与えない
bit 1	FSIM9: COGx 立ち下がリイベント入力 9 モードビット FIS9 = 1: 1 = PWM5 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = PWM5 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS9 = 0: PWM5 出力は立ち下がリイベントに影響を与えない
bit 0	FSIM8: COGx 立ち下がリイベント入力 8 モードビット FIS8 = 1: 1 = PWM4 出力が High から Low へ遷移すると、立ち下がリイベント位相遅延の後、立ち下がリイベントを発生させる 0 = PWM4 出力の Low レベルでただちに立ち下がリイベントを発生させる FIS8 = 0: PWM4 出力は立ち下がリイベントに影響を与えない

レジスタ 27-11: COGxASD0: COG 自動シャットダウン制御レジスタ 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7 **ASE:** 自動シャットダウン イベント ステータスビット

1 = COG がシャットダウン状態である

0 = COG はシャットダウン状態でないか、次の立ち上がりイベントでシャットダウン状態を終了する

bit 6 **ARSEN:** 自動再起動イネーブルビット

1 = 自動再起動を有効にする

0 = 自動再起動を無効にする

bit 5-4 **ASDBD<1:0>:** COGxB および COGxD 自動シャットダウン オーバーライド レベル選択ビット

11 = シャットダウンが有効な間、COGxB と COGxD に論理「1」を駆動する

10 = シャットダウンが有効な間、COGxB と COGxD に論理「0」を駆動する

01 = シャットダウンが有効な間、COGxB と COGxD を 3 ステートにする

00 = シャットダウンが有効な間、ピンの非アクティブ状態 (極性を含む) を COGxB と COGxD に駆動する

bit 3-2 **ASDAC<1:0>:** COGxA および COGxC 自動シャットダウン オーバーライド レベル選択ビット

11 = シャットダウンが有効な間、COGxA と COGxC に論理「1」を駆動する

10 = シャットダウンが有効な間、COGxA と COGxC に論理「0」を駆動する

01 = シャットダウンが有効な間、COGxA と COGxC を 3 ステートにする

00 = シャットダウンが有効な間、ピンの非アクティブ状態 (極性を含む) を COGxA と COGxC に駆動する

bit 1-0 **未実装:** 「0」として読み出し

PIC16(L)F1764/5/8/9

レジスタ 27-12: COGxASD1: COG 自動シャットダウン制御レジスタ 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

- bit 7 **AS7E:** COGx 自動シャットダウン要因イネーブルビット 7
1 = Timer4_output が High になると COGx をシャットダウンする
0 = Timer4_output はシャットダウンに影響を与えない
- bit 6 **AS6E:** COGx 自動シャットダウン要因イネーブルビット 6
1 = Timer2_output が High になると COGx をシャットダウンする
0 = Timer2_output はシャットダウンに影響を与えない
- bit 5 **AS5E:** COGx 自動シャットダウン要因イネーブルビット 5
1 = CLC LC2_out が Low になると COGx をシャットダウンする
0 = CLC2 出力はシャットダウンに影響を与えない
- bit 4 **AS4E:** COGx 自動シャットダウン要因イネーブルビット 4
1 = コンパレータ sync_C4OUT が Low になると COGx をシャットダウンする
0 = コンパレータ 4 出力はシャットダウンに影響を与えない
- bit 3 **AS3E:** COGx 自動シャットダウン要因イネーブルビット 3
1 = コンパレータ sync_C3OUT が Low になると COGx をシャットダウンする
0 = コンパレータ 3 出力はシャットダウンに影響を与えない
- bit 2 **AS2E:** COGx 自動シャットダウン要因イネーブルビット 2
1 = コンパレータ sync_C2OUT が Low になると COGx をシャットダウンする
0 = コンパレータ 2 出力はシャットダウンに影響を与えない
- bit 1 **AS1E:** COGx 自動シャットダウン要因イネーブルビット 1
1 = コンパレータ sync_C1OUT が Low になると COGx をシャットダウンする
0 = コンパレータ 1 出力はシャットダウンに影響を与えない
- bit 0 **AS0E:** COGx 自動シャットダウン要因イネーブルビット 0
1 = COGxINPPS 制御レジスタで選択したピンが Low になると COGx をシャットダウンする
0 = COGxINPPS 制御レジスタで選択したピンはシャットダウンに影響を与えない

レジスタ 27-13: COGxSTR: COG ステアリング制御レジスタ 1⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7	SDATD: COGxD 固定出力データビット 1 = COGxD の固定データを High にする 0 = COGxD の固定データを Low にする
bit 6	SDATC: COGxC 固定出力データビット 1 = COGxC の固定データを High にする 0 = COGxC の固定データを Low にする
bit 5	SDATB: COGxB 固定出力データビット 1 = COGxB の固定データを High にする 0 = COGxB の固定データを Low にする
bit 4	SDATA: COGxA 固定出力データビット 1 = COGxA の固定データを High にする 0 = COGxA の固定データを Low にする
bit 3	STRD: COGxD ステアリング制御ビット 1 = COGxD の出力を COGxD 波形とし、POLD ビットで極性を制御する 0 = COGxD の出力を SDATD ビットで設定した固定データレベルとする
bit 2	STRC: COGxC ステアリング制御ビット 1 = COGxC の出力を COGxC 波形とし、POLC ビットで極性を制御する 0 = COGxC の出力を SDATC ビットで設定した固定データレベルとする
bit 1	STRB: COGxB ステアリング制御ビット 1 = COGxB の出力を COGxB 波形とし、POLB ビットで極性を制御する 0 = COGxB の出力を SDATB ビットで設定した固定データレベルとする
bit 0	STRA: COGxA ステアリング制御ビット 1 = COGxA の出力を COGxA 波形とし、POLA ビットで極性を制御する 0 = COGxA の出力を SDATA ビットで設定した固定データレベルとする

Note 1: COGxCON0 レジスタの MD<1:0> ビット = 00x の時のみステアリングはアクティブです ([レジスタ 27-1](#) 参照)。

PIC16(L)F1764/5/8/9

レジスタ 27-14: COGxDBR: COG 立ち上がりイベント デッドバンド カウントレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
		DBR<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **DBR<5:0>:** 立ち上がりイベント デッドバンド カウント値ビット

RDBS = 0:

= 立ち上がりイベントの後、主出力が変化するまでの遅延を COGx クロック周期の数で指定する

RDBS = 1:

= 立ち上がりイベントの後、主出力が変化するまでの遅延を遅延チェーンの素子数で指定する

レジスタ 27-15: COGxDBF: COG 立ち下がりイベント デッドバンド カウントレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
		DBF<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **DBF<5:0>:** 立ち下がりイベント デッドバンド カウント値ビット

FDBS = 0:

= 立ち下がりイベント入力の後、相補副出力が変化するまでの遅延を COGx クロック周期の数で指定する

FDBS = 1:

= 立ち下がりイベント入力の後、相補副出力が変化するまでの遅延を遅延チェーンの素子数で指定する

レジスタ 27-16: COGxBLKR: COG 立ち上がりイベント ブランキング カウントレジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
		BLKR<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **BLKR<5:0>:** 立ち上がりイベント ブランキング カウント値ビット

= 立ち下がりイベント入力を禁止する期間を COGx クロック周期の数で指定する

PIC16(L)F1764/5/8/9

レジスタ 27-17: COGxBLKF: COG 立ち下がリイベントブランキング カウントレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
		BLKF<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **BLKF<5:0>:** 立ち下がリイベントブランキング カウント値ビット
= 立ち上がりイベント入力を禁止する期間を COGx クロック周期の数で指定する

レジスタ 27-18: COGxPHR: COG 立ち上がりイベント位相遅延カウントレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	PHR<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **PHR<5:0>:** 立ち上がりイベント位相遅延カウント値ビット
= 立ち上がりイベントの遅延を COGx クロック周期の数で指定する

レジスタ 27-19: COGxPHF: COG 立ち下がリイベント位相遅延カウントレジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	PHF<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-6 **未実装:** 「0」として読み出し

bit 5-0 **PHF<5:0>:** 立ち下がリイベント位相遅延カウント値ビット
= 立ち下がリイベントの遅延を COGx クロック周期の数で指定する

PIC16(L)F1764/5/8/9

表 27-4: COGx 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB6	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC0	ANSC0	151
COGxASD0	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—	325
COGxASD1	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	326
COGxBLKR	—	—	BLKR<5:0>						328
COGxBLKF	—	—	BLKF<5:0>						329
COGxCON0	EN	LD	—	CS<1:0>		MD<2:0>			315
COGxCON1	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	316
COGxDBR	—	—	DBR<5:0>						328
COGxDBF	—	—	DBF<5:0>						328
COGxFIS0	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	321
COGxFIS1	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	322
COGxFSIM0	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	323
COGxFSIM1	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	324
COGxPHR	—	—	PHR<5:0>						329
COGxPHF	—	—	PHF<5:0>						329
COGxPPS	—	—	—	COG1PPS<4:0>					158、160
COGxRIS0	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	317
COGxRIS1	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	318
COGxRSIM0	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	319
COGxRSIM1	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	320
COGxSTR	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	327
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
RxyPPS	—	—	—	RxyPPS<4:0>					158

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し。網掛けの部分は COG では使いません。

Note 1: PIC16(L)F1768/9 のみです。

28.0 構成可能なロジックセル (CLC)

構成可能なロジックセル (CLCx) は、ソフトウェア実行速度の制約を受けずに動作する、プログラム可能なロジック機能を提供します。ロジックセルには最大 32 の入力信号を接続でき、構成可能なゲートによってこれらの信号を 4 本のロジックラインに集約します。このラインによって、8 つの選択可能な単一出カロジック機能の 1 つを駆動します。

入力信号は以下の組み合わせです。

- I/O ピン
- 内部クロック
- 周辺モジュール
- レジスタビット

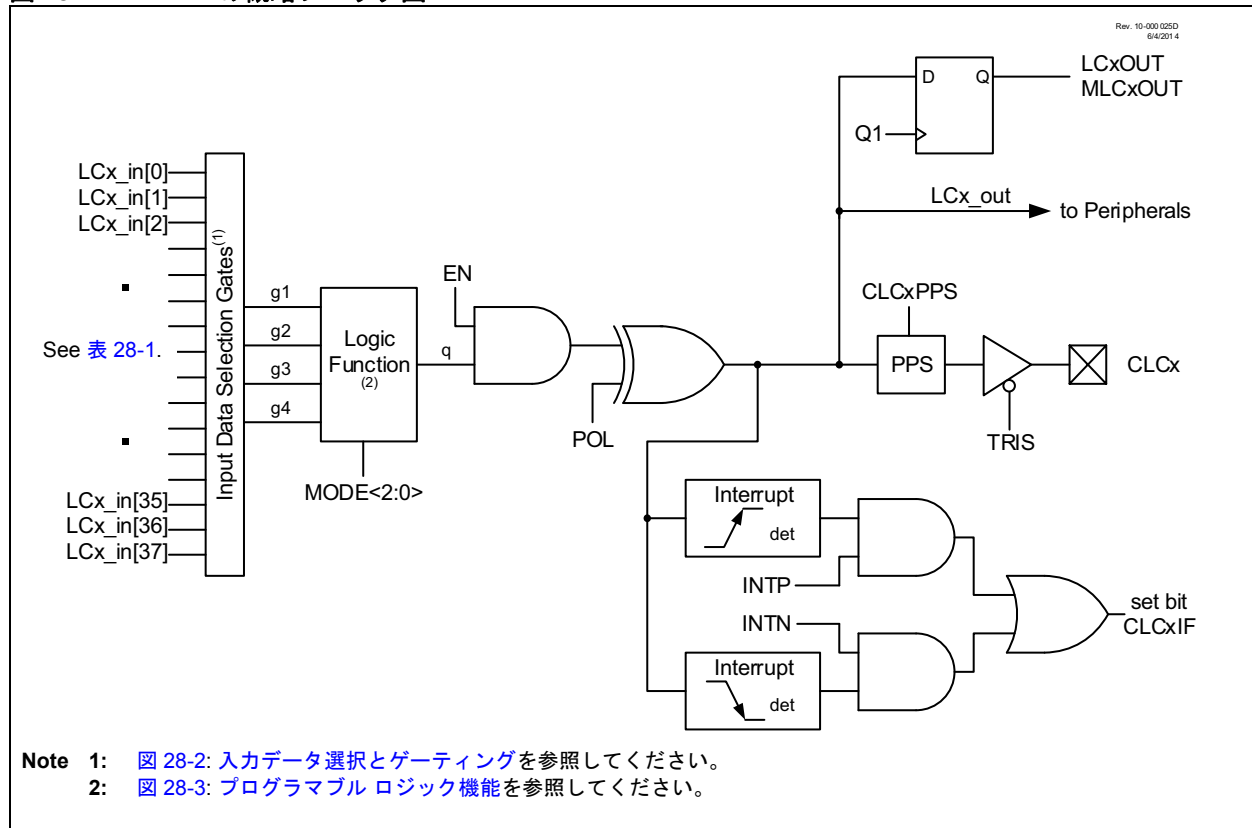
出力は、内部周辺モジュールと外部出力ピンに接続できます。

CLCx 内の信号の流れは図 28-1 の概略ブロック図を参照してください。

以下のような構成が可能です。

- 組み合わせロジック
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- ラッチ
 - S-R ラッチ
 - セット/リセット付き同期 D ラッチ
 - セット/リセット付き透過 D ラッチ
 - リセット付き同期 J-K ラッチ

図 28-1: CLCx の概略ブロック図



PIC16(L)F1764/5/8/9

28.1 CLCx の設定

CLCx モジュールは、ロジック信号フローの 4 つの段を設定してプログラムします。4 つの段とは以下の通りです。

- データの選択
- データのゲーティング
- ロジック機能の選択
- 出力極性の選択

各段は、対応する CLCx 特殊機能レジスタに書き込む事で実行時に設定します。この方式は、プログラム実行中に動作を停止せずにロジックを再構成できるという利点があります。

28.1.1 データの選択

構成可能ロジックへの入力として 32 の信号が使えます。4 つの 32 入力マルチプレクサによって、次段に通過させる入力を選択します。

データの選択は、図 28-2 の左側に示した 4 つのマルチプレクサで実行します。この図では、データ入力を通し番号による総称入力名で表記しています。

表 28-1 に、総称入力名と各 CLC モジュールの実際の信号の対応を示します。選択したデータ入力に対応する MUX 選択コードを示しています。DxS は、それぞれ MUX 選択入力コード D1S<4:0> ~ D4S<4:0> の略号です。

データ入力は、CLCxSEL0 ~ CLCxSEL3 レジスタ (レジスタ 28-3 ~ レジスタ 28-6) で選択します。

Note: 電源投入時、データ選択は未定義の状態です。

表 28-1: CLCx データ入力選択

データ入力	dy DxS	CLCx
LCx_in[37]	100101	Fosc
LCx_in[36]	100100	HFINTOSC
LCx_in[35]	100011	LFINTOSC
LCx_in[34]	100010	FRC (ADC RC クロック)
LCx_in[33]	100001	IOCIF セット
LCx_in[32]	100000	Timer6_postscaled
LCx_in[31]	011111	Timer4_postscaled
LCx_in[30]	011110	Timer2_postscaled
LCx_in[29]	011101	Timer5 オーバーフロー
LCx_in[28]	011100	Timer3 オーバーフロー
LCx_in[27]	011011	Timer1 オーバーフロー
LCx_in[26]	011010	Timer0 オーバーフロー
LCx_in[25]	011001	EUSART 受信
LCx_in[24]	011000	EUSART 送信
LCx_in[23]	010111	ZCD1_output
LCx_in[22]	010110	MSSP1 の SDO/SDA
LCx_in[21]	010101	MSSP1 の SCL/SCK
LCx_in[20]	010100	PWM6_out
LCx_in[19]	010011	PWM5_out
LCx_in[18]	010010	PWM4_out
LCx_in[17]	010001	PWM3_out
LCx_in[16]	010000	CCP2_out
LCx_in[15]	001111	CCP1_out
LCx_in[14]	001110	COG2B
LCx_in[13]	001101	COG2A
LCx_in[12]	001100	COG1B
LCx_in[11]	001011	COG1A
LCx_in[10]	001010	sync_C4OUT
LCx_in[9]	001001	sync_C3OUT
LCx_in[8]	001000	sync_C2OUT
LCx_in[7]	000111	sync_C1OUT
LCx_in[6]	000110	CLC3 からの LC3_out
LCx_in[5]	000101	CLC2 からの LC2_out
LCx_in[4]	000100	CLC1 からの LC1_out
LCx_in[3]	000011	CLCIN3PPS レジスタで選択した CLCIN3 ピン入力
LCx_in[2]	000010	CLCIN2PPS レジスタで選択した CLCIN2 ピン入力
LCx_in[1]	000001	CLCIN1PPS レジスタで選択した CLCIN1 ピン入力
LCx_in[0]	000000	CLCIN0PPS レジスタで選択した CLCIN0 ピン入力

28.1.2 データのゲーティング

入力マルチプレクサの出力は、データのゲーティング段を通して選択したロジック機能に入力されます。各データゲートは、選択した4つの入力の任意の組み合わせを送出できます。

Note: 電源投入時、データのゲーティングは未定義の状態です。

ゲート段の機能は、単なる信号の送出力にとどまりません。ゲートは、各入力信号を反転または非反転データとして送出力するよう設定できます。送出力された信号は、各データゲートで互いに AND 演算されます。各ゲートの出力は、ロジック機能段に入力する前に反転できます。

ゲーティング回路は本質的には1～4入力の AND/NAND/OR/NOR ゲートです。全ての入力が反転信号で、出力も反転する場合、ゲートは有効な全てのデータ入力の論理和 (OR) になります。入力と出力を反転しない場合、ゲートは有効な全てのデータ入力の論理積 (AND) になります。

表 28-2 に、ゲートロジック選択ビットによってゲート1から得られる基本ロジックを示します。表は4入力変数のロジックを示していますが、各ゲートで3入力以下を使う設定も可能です。入力を1つも選択しない場合、出力はゲート出力極性ビットに応じて0または1になります。

表 28-2: データゲートロジック

CLCxGLS0	G1POL	ゲートロジック
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	論理 0
0x00	1	論理 1

1つの入力信号源からの非反転値と反転値の両方を選択する事も可能ですが、そのような設定は推奨しません。両方を選択した場合のゲート出力は他の入力にかかわらず0ですが、ロジックグリッチ(過渡動作によるパルス)が発生する可能性があります。チャンネルの出力を0または1にする必要がある場合、全てのゲートビットを0にセットしてゲート極性ビットを目的のレベルとなるように設定する方法を推奨します。

データのゲーティングは、ロジックゲート選択レジスタによって、以下のように設定します。

- ゲート 1: CLCxGLS0([レジスタ 28-7](#))
- ゲート 2: CLCxGLS1([レジスタ 28-8](#))
- ゲート 3: CLCxGLS2([レジスタ 28-9](#))
- ゲート 4: CLCxGLS3([レジスタ 28-10](#))

レジスタ番号の接尾辞はゲート番号とは異なります。これは、他のデバイスが同じレジスタで複数のゲートを選択する場合があるためです。

図 28-2 の右側に、データのゲーティングを示します。1つのゲートのみ詳細を示しています。残りの3つのゲートも、そのゲートを有効にするデータイネーブル信号を除いて同じ構成です。

28.1.3 ロジック機能

以下の8つのロジック機能が使えます。

- AND-OR
- OR-XOR
- AND
- S-R ラッチ
- D型フリップフロップ(セット/リセット入力付き)
- D型フリップフロップ(リセット入力付き)
- JK型フリップフロップ(リセット入力付き)
- トランスペアレントラッチ(セット/リセット入力付き)

これらのロジック機能については[図28-3](#)を参照してください。各ロジック機能は4入力/1出力です。4つの入力は、前段の4つのデータゲートの出力です。出力は反転段に入力され、そこから他の周辺モジュール、出力ピンに送出力するか、CLCx自体に戻されます。

28.1.4 出力極性

構成可能なロジックセルの最終段は出力極性段です。CLCxCONレジスタのPOLビットをセットすると、ロジック段からの出力信号が反転します。割り込みを有効にした状態で極性を変更すると、出力の遷移によって割り込みが発生します。

28.1.5 CLCx の設定ステップ

CLCx を設定するには、以下のステップを実行します。

- EN ビットをクリアして CLCx を無効にする。
- CLCxSEL0 ~ CLCxSEL3 レジスタで入力を選択する (表 28-1 参照)。
- 関連する ANSEL ビットを全てクリアする。
- 入力に関連する TRIS ビットを全てセットする。
- 出力に関連する TRIS ビットを全てクリアする。
- CLCxGLS0、CLCxGLS1、CLCxGLS2、CLCxGLS3 レジスタで、4 つのゲートを通してよう選択した入力を有効にする。
- CLCxPOL レジスタの POLy ビットでゲート出力の極性を選択する。
- CLCxCON レジスタの MODE<2:0> ビットでロジック機能を選択する。
- CLCxPOL レジスタの POL ビットでロジック出力極性を選択する (このステップは、前述のゲート出力極性のステップと一緒に実行する事も可能)。
- デバイスピンを駆動する場合、駆動するピンの PPS 制御レジスタを設定し、その出力に対応する TRIS ビットをクリアする。
- 割り込みが必要な場合、以下のビットで設定する。
 - 立ち上がりエッジで割り込みを発生させる場合、CLCxCON レジスタの INTP ビットをセットする。
 - 立ち下がりエッジで割り込みを発生させる場合、CLCxCON レジスタの INTN ビットをセットする。
 - 対応する PIE レジスタの CLCxIE ビットをセットする。
 - INTCON レジスタの GIE ビットと PEIE ビットをセットする。
- CLCxCON レジスタの EN ビットをセットして CLCx を有効にする。

28.2 CLCx 割り込み

CLCx の出力値が変化すると、セットした割り込みイネーブルビットの種類に応じて割り込みが生成されます。この目的のために、各 CLC は立ち上がりエッジの検出機能と立ち下がりエッジの検出機能を備えています。

どちらかのエッジ検出機能がトリガされ、対応するイネーブルビットがセットされている場合、対応する PIR レジスタの CLCxIF ビットがセットされます。INTP ビットは立ち上がりエッジ割り込みを、INTN ビットは立ち下がりエッジ割り込みを有効にします。いずれのビットも CLCxCON レジスタに含まれます。

この割り込みを完全に有効にするには、以下のビットをセットします。

- CLCxCON レジスタの ON ビット
- 対応する PIE レジスタの CLCxIE ビット
- CLCxCON レジスタの INTP ビット (立ち上がりエッジ検出の場合)
- CLCxCON レジスタの INTN ビット (立ち下がりエッジ検出の場合)
- INTCON レジスタの PEIE および GIE ビット

対応する PIR レジスタの CLCxIF ビットは、割り込みサービスの一部としてソフトウェアでクリアする必要があります。このフラグをクリア中に次のエッジが検出された場合、シーケンス完了時にフラグはセットされたままです。

28.3 出力のミラーコピー

LCxCON 出力ビットは全て、CLCDATA レジスタにミラーコピーがあります。このレジスタを読み出すと、全 CLC 出力を同時に読み出せます。この方法を使えば、CLCxCON レジスタの CLCxOUT ビットを別々にテストまたは読み出した場合に発生する読み出しスキューを防ぐことができます。

28.4 リセットの影響

CLCxCON レジスタはリセットによって 0 にクリアされます。その他全ての選択とゲーティングの値は変化しません。

28.5 スリープ中の動作

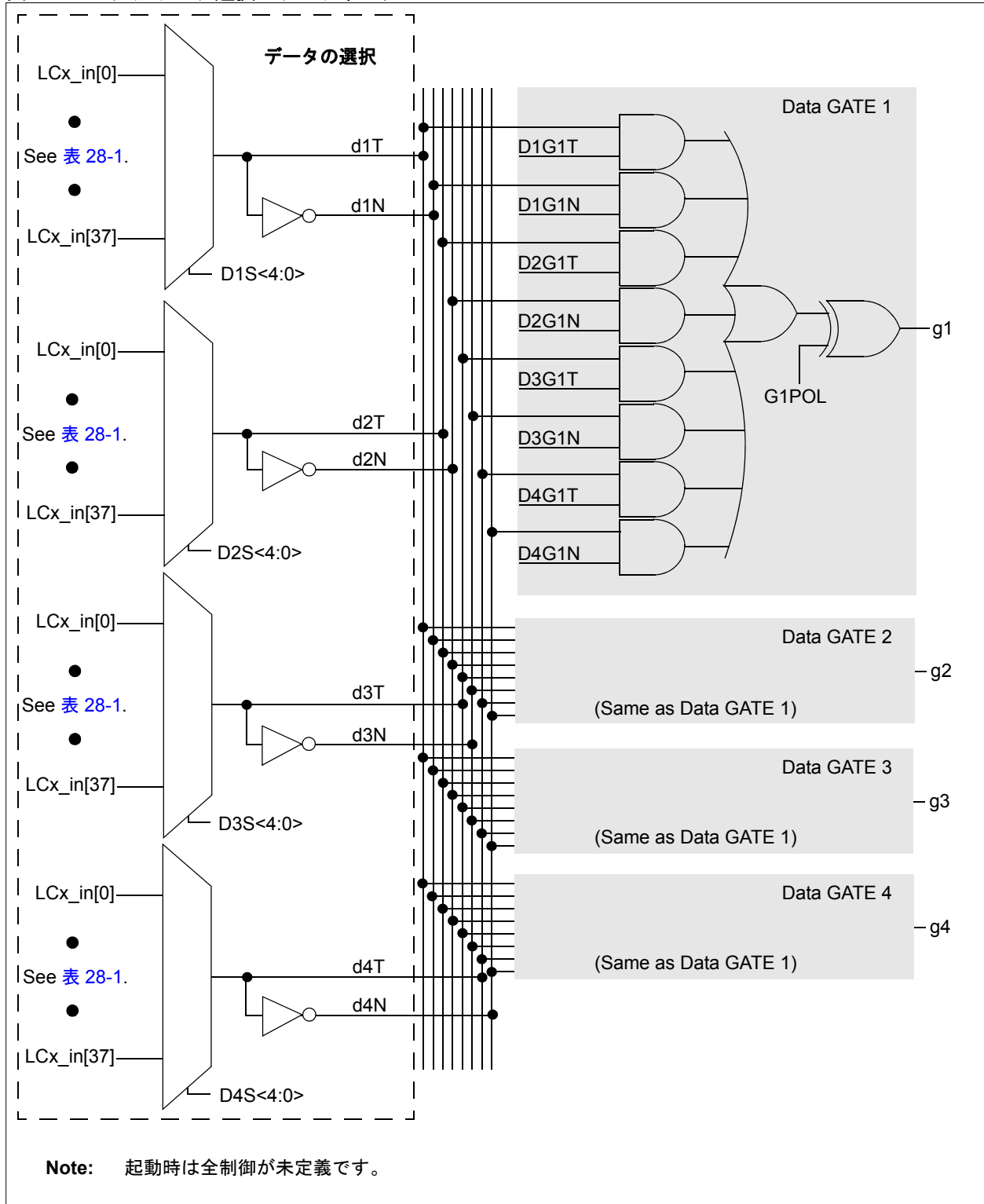
CLC モジュールはシステムクロックから独立して動作します。選択されている入力信号がアクティブであれば、スリープ中も動作を継続します。

CLC モジュールが有効で、入力信号として HFINTOSC を選択している場合、システムクロック源にかかわらず、HFINTOSC はスリープ中もアクティブ状態を維持します。

言い換えれば、システムクロックと CLC 入力信号の両方に HFINTOSC を選択しており、CLC が有効の場合、CPU はスリープ中にアイドル状態に移行しますが、CLC は動作を継続して HFINTOSC はアクティブ状態を維持します。

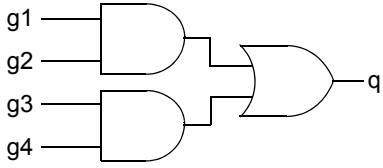
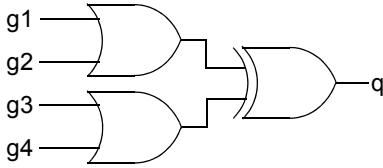
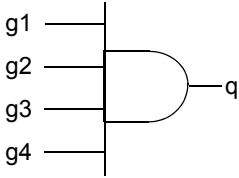
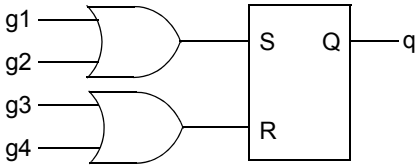
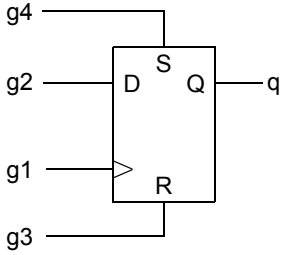
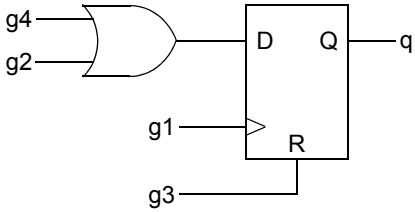
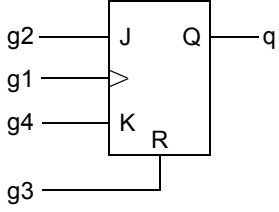
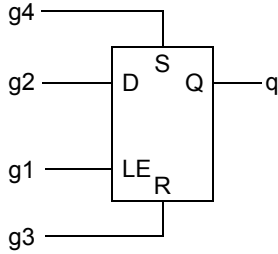
その場合、スリープ中の消費電流に大きく影響します。

図 28-2: 入力データ選択とゲーティング



PIC16(L)F1764/5/8/9

図 28-3: プログラマブル ロジック機能

<p style="text-align: center;">AND – OR</p>  <p style="text-align: center;">MODE<2:0> = 000</p>	<p style="text-align: center;">OR – XOR</p>  <p style="text-align: center;">MODE<2:0> = 001</p>
<p style="text-align: center;">4-Input AND</p>  <p style="text-align: center;">MODE<2:0> = 010</p>	<p style="text-align: center;">S-R Latch</p>  <p style="text-align: center;">MODE<2:0> = 011</p>
<p style="text-align: center;">1-Input D Flip-Flop with S and R</p>  <p style="text-align: center;">MODE<2:0> = 100</p>	<p style="text-align: center;">2-Input D Flip-Flop with R</p>  <p style="text-align: center;">MODE<2:0> = 101</p>
<p style="text-align: center;">J-K Flip-Flop with R</p>  <p style="text-align: center;">MODE<2:0> = 110</p>	<p style="text-align: center;">1-Input Transparent Latch with S and R</p>  <p style="text-align: center;">MODE<2:0> = 111</p>

28.6 レジスタ定義 : CLC 制御

表 28-3 に CLC 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1 「レジスタおよびビット命名法」](#) を参照してください。

表 28-3:

周辺モジュール	ビット名の接頭辞
CLC1	LC1
CLC2	LC2

レジスタ 28-1: **CLCxCON: 構成可能なロジックセル (CLC) 制御レジスタ**

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	INTP	INTN	MODE<2:0>		
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7 **EN:** 構成可能なロジックセル イネーブルビット
 1 = 構成可能なロジックセルを有効にして入力信号の論理演算値を出力する
 0 = 構成可能なロジックセルを無効にする (出力は論理「0」)
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **OUT:** 構成可能なロジックセル データ出力ビット
 読み出し専用: POL による操作後のロジックセル出力データ (lcx_out 配線よりサンプリング)
- bit 4 **INTP:** 構成可能なロジックセル立ち上がりエッジ割り込みイネーブルビット
 1 = lcx_out に立ち上がりエッジが発生した場合に CLCxIF をセットする
 0 = CLCxIF をセットしない
- bit 3 **INTN:** 構成可能なロジックセル立ち下がりエッジ割り込みイネーブルビット
 1 = lcx_out に立ち下がりエッジが発生した場合に CLCxIF をセットする
 0 = CLCxIF をセットしない
- bit 2-0 **MODE<2:0>:** 構成可能なロジックセル機能モードビット
 111 = セルをセット/リセット付き 1 入力透過ラッチに設定する
 110 = セルをリセット付き J-K フリップフロップに設定する
 101 = セルをリセット付き 2 入力 D フリップフロップに設定する
 100 = セルをセット/リセット付き 1 入力 D フリップフロップに設定する
 011 = セルをセット/リセットラッチに設定する
 010 = セルを 4 入力 AND に設定する
 001 = セルを OR-XOR に設定する
 000 = セルを AND-OR に設定する

PIC16(L)F1764/5/8/9

レジスタ 28-2: CLCxPOL: 信号極性制御レジスタ

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
POL	—	—	—	G4POL	G3POL	G2POL	G1POL
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **POL:** LCOOUT 極性制御ビット
1 = ロジックセルの出力を反転する
0 = ロジックセルの出力を反転しない
- bit 6-4 **未実装:** 「0」として読み出し
- bit 3 **G4POL:** ゲート 4 出力極性制御ビット
1 = ロジックセルへの入力時にゲート 4 の出力を反転する
0 = ゲート 4 の出力を反転しない
- bit 2 **G3POL:** ゲート 3 出力極性制御ビット
1 = ロジックセルへの入力時にゲート 3 の出力を反転する
0 = ゲート 3 の出力を反転しない
- bit 1 **G2POL:** ゲート 2 出力極性制御ビット
1 = ロジックセルへの入力時にゲート 2 の出力を反転する
0 = ゲート 2 の出力を反転しない
- bit 0 **G1POL:** ゲート 1 出力極性制御ビット
1 = ロジックセルへの入力時にゲート 1 の出力を反転する
0 = ゲート 1 の出力を反転しない

レジスタ 28-3: CLCxSEL0: CLCx データ 1 選択レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D1S<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **D1S<5:0>:** CLCx データ 1 入力選択ビット
[表 28-1 参照](#)

レジスタ 28-4: CLCxSEL1: 総称 CLCx データ 2 選択レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D2S<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **D2S<5:0>:** CLCx データ 2 入力選択ビット
[表 28-1 参照](#)

レジスタ 28-5: CLCxSEL2: 総称 CLCx データ 3 選択レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D3S<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **D3S<5:0>:** CLCx データ 3 入力選択ビット
[表 28-1 参照](#)

PIC16(L)F1764/5/8/9

レジスタ 28-6: CLCxSEL3: 総称 CLCx データ 4 選択レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D4S<5:0>					
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-6 **未実装:** 「0」として読み出し
bit 5-0 **D4S<5:0>:** CLCx データ 4 入力選択ビット
 [表 28-1 参照](#)

レジスタ 28-7: CLCxGLS0: ゲート 1 ロジック選択レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7 **G1D4T:** ゲート 1 データ 4 True(非反転)ビット
 1 = d4T を g1 にゲーティングする
 0 = d4T を g1 にゲーティングしない
bit 6 **G1D4N:** ゲート 1 データ 4 Negated(反転)ビット
 1 = d4N を g1 にゲーティングする
 0 = d4N を g1 にゲーティングしない
bit 5 **G1D3T:** ゲート 1 データ 3 True(非反転)ビット
 1 = d3T を g1 にゲーティングする
 0 = d3T を g1 にゲーティングしない
bit 4 **G1D3N:** ゲート 1 データ 3 Negated(反転)ビット
 1 = d3N を g1 にゲーティングする
 0 = d3N を g1 にゲーティングしない
bit 3 **G1D2T:** ゲート 1 データ 2 True(非反転)ビット
 1 = d2T を g1 にゲーティングする
 0 = d2T を g1 にゲーティングしない
bit 2 **G1D2N:** ゲート 1 データ 2 Negated(反転)ビット
 1 = d2N を g1 にゲーティングする
 0 = d2N を g1 にゲーティングしない
bit 1 **G1D1T:** ゲート 1 データ 1 True(非反転)ビット
 1 = d1T を g1 にゲーティングする
 0 = d1T を g1 にゲーティングしない
bit 0 **G1D1N:** ゲート 1 データ 1 Negated(反転)ビット
 1 = d1N を g1 にゲーティングする
 0 = d1N を g1 にゲーティングしない

レジスタ 28-8: CLCxGLS1: ゲート 2 ロジック選択レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7 **G2D4T:** ゲート 2 データ 4 True(非反転) ビット
 1 = d4T を g2 にゲーティングする
 0 = d4T を g2 にゲーティングしない
- bit 6 **G2D4N:** ゲート 2 データ 4 Negated(反転) ビット
 1 = d4N を g2 にゲーティングする
 0 = d4N を g2 にゲーティングしない
- bit 5 **G2D3T:** ゲート 2 データ 3 True(非反転) ビット
 1 = d3T を g2 にゲーティングする
 0 = d3T を g2 にゲーティングしない
- bit 4 **G2D3N:** ゲート 2 データ 3 Negated(反転) ビット
 1 = d3N を g2 にゲーティングする
 0 = d3N を g2 にゲーティングしない
- bit 3 **G2D2T:** ゲート 2 データ 2 True(非反転) ビット
 1 = d2T を g2 にゲーティングする
 0 = d2T を g2 にゲーティングしない
- bit 2 **G2D2N:** ゲート 2 データ 2 Negated(反転) ビット
 1 = d2N を g2 にゲーティングする
 0 = d2N を g2 にゲーティングしない
- bit 1 **G2D1T:** ゲート 2 データ 1 True(非反転) ビット
 1 = d1T を g2 にゲーティングする
 0 = d1T を g2 にゲーティングしない
- bit 0 **G2D1N:** ゲート 2 データ 1 Negated(反転) ビット
 1 = d1N を g2 にゲーティングする
 0 = d1N を g2 にゲーティングしない

PIC16(L)F1764/5/8/9

レジスタ 28-9: CLCxGLS2: ゲート 3 ロジック選択レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **G3D4T:** ゲート 3 データ 4 True(非反転)ビット
1 = d4T を g3 にゲーティングする
0 = d4T を g3 にゲーティングしない
- bit 6 **G3D4N:** ゲート 3 データ 4 Negated(反転)ビット
1 = d4N を g3 にゲーティングする
0 = d4N を g3 にゲーティングしない
- bit 5 **G3D3T:** ゲート 3 データ 3 True(非反転)ビット
1 = d3T を g3 にゲーティングする
0 = d3T を g3 にゲーティングしない
- bit 4 **G3D3N:** ゲート 3 データ 3 Negated(反転)ビット
1 = d3N を g3 にゲーティングする
0 = d3N を g3 にゲーティングしない
- bit 3 **G3D2T:** ゲート 3 データ 2 True(非反転)ビット
1 = d2T を g3 にゲーティングする
0 = d2T を g3 にゲーティングしない
- bit 2 **G3D2N:** ゲート 3 データ 2 Negated(反転)ビット
1 = d2N を g3 にゲーティングする
0 = d2N を g3 にゲーティングしない
- bit 1 **G3D1T:** ゲート 3 データ 1 True(非反転)ビット
1 = d1T を g3 にゲーティングする
0 = d1T を g3 にゲーティングしない
- bit 0 **G3D1N:** ゲート 3 データ 1 Negated(反転)ビット
1 = d1N を g3 にゲーティングする
0 = d1N を g3 にゲーティングしない

レジスタ 28-10: CLCxGLS3: ゲート 4 ロジック選択レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **G4D4T:** ゲート 4 データ 4 True(非反転) ビット
1 = d4T を g4 にゲーティングする
0 = d4T を g4 にゲーティングしない
- bit 6 **G4D4N:** ゲート 4 データ 4 Negated(反転) ビット
1 = d4N を g4 にゲーティングする
0 = d4N を g4 にゲーティングしない
- bit 5 **G4D3T:** ゲート 4 データ 3 True(非反転) ビット
1 = d3T を g4 にゲーティングする
0 = d3T を g4 にゲーティングしない
- bit 4 **G4D3N:** ゲート 4 データ 3 Negated(反転) ビット
1 = d3N を g4 にゲーティングする
0 = d3N を g4 にゲーティングしない
- bit 3 **G4D2T:** ゲート 4 データ 2 True(非反転) ビット
1 = d2T を g4 にゲーティングする
0 = d2T を g4 にゲーティングしない
- bit 2 **G4D2N:** ゲート 4 データ 2 Negated(反転) ビット
1 = d2N を g4 にゲーティングする
0 = d2N を g4 にゲーティングしない
- bit 1 **G4D1T:** ゲート 4 データ 1 True(非反転) ビット
1 = d1T を g4 にゲーティングする
0 = d1T を g4 にゲーティングしない
- bit 0 **G4D1N:** ゲート 4 データ 1 Negated(反転) ビット
1 = d1N を g4 にゲーティングする
0 = d1N を g4 にゲーティングしない

PIC16(L)F1764/5/8/9

レジスタ 28-11: CLCDATA: CLC データ出力

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-3 **未実装:** 「0」として読み出し
bit 2 MLC3OUT: LC3OUT ビットのミラーコピー
bit 1 MLC2OUT: LC2OUT ビットのミラーコピー
bit 0 MLC1OUT: LC1OUT ビットのミラーコピー

表 28-4: CLCx 関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
CLCxCON	EN	—	OUT	INTP	INTN	MODE<2:0>			337
CLCDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	344
CLCxGLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	340
CLCxGLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	341
CLCxGLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	342
CLCxGLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	343
CLCxPOL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	338
CLCxSEL0	—	—	D1S<5:0>						339
CLCxSEL1	—	—	D2S<5:0>						339
CLCxSEL2	—	—	D3S<5:0>						339
CLCxSEL3	—	—	D4S<5:0>						340
CLCxPPS	—	—	—	CLCxPPS<4:0>					158、160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	102
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
RxyPPS	—	—	—	RxyPPS<4:0>					158
TRISA	—	—	TRISA5	TRISA4	— ⁽³⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例: — = 未実装、「0」として読み出し。網掛けの部分は CLC モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

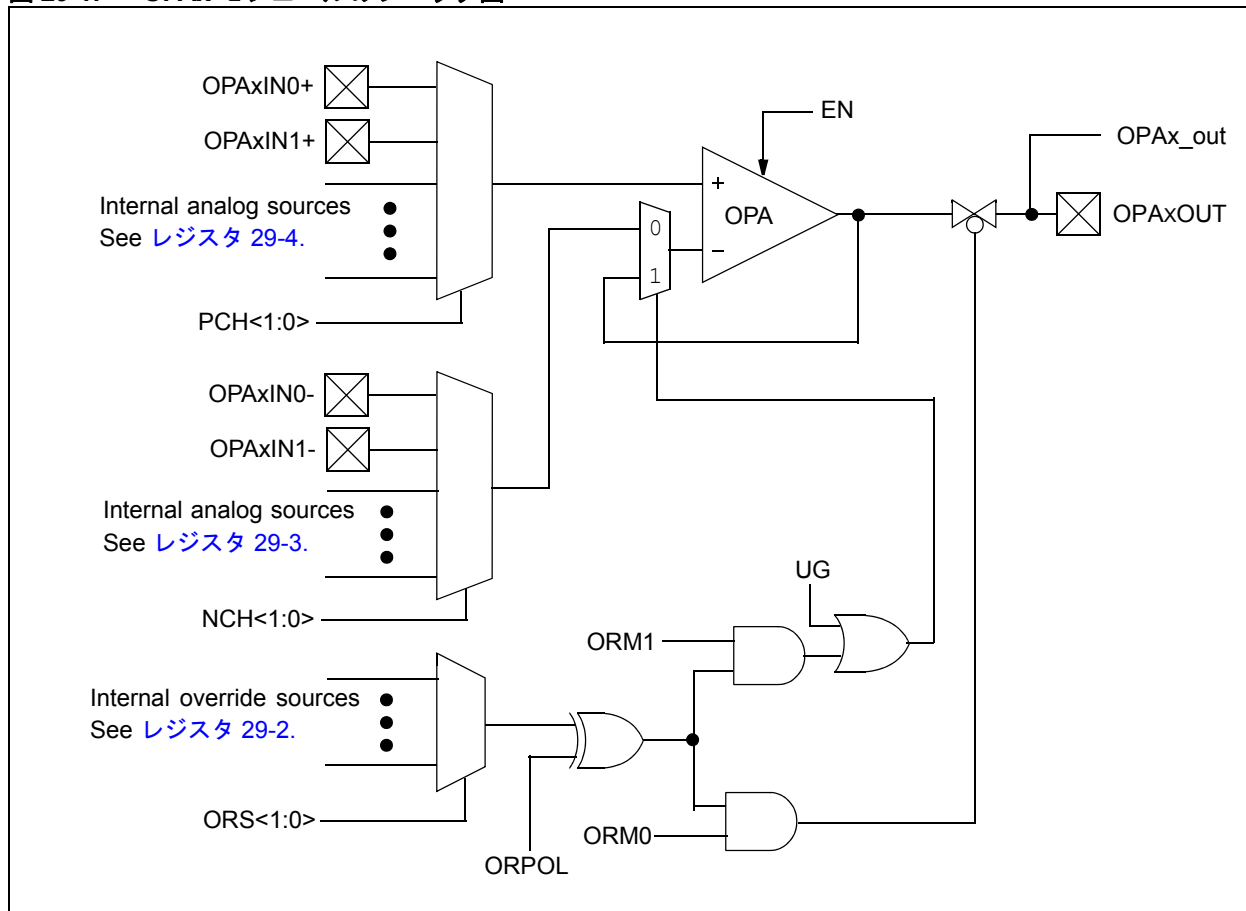
PIC16(L)F1764/5/8/9

29.0 オペアンプ (OPA) モジュール

本オペアンプ (OPA) モジュールは標準的な 3 端子デバイスで、動作には外部帰還が必要です。OPA モジュールには以下の特長があります。

- I/O ポート経由で外部接続
- リーク電流が少ない
- 工場で入力オフセット電圧を校正済み
- ユニティゲイン制御
- プログラマブルな正負入力源選択
- オーバーライド制御
 - 強制 3 ステート出力
 - 強制ユニティゲイン

図 29-1: OPAx モジュールのブロック図



29.1 OPA モジュールの性能

OPA モジュールの一般的な AC および DC 特性仕様には以下のものがあります。

- コモンモード電圧レンジ
- リーク電流
- 入力オフセット電圧
- 開ループゲイン
- ゲイン帯域幅積

コモンモード電圧レンジは、OPA+/OPA- 入力の仕様電圧レンジです。OPA モジュールはこの仕様内で動作します。OPA モジュールは、VSS ~ VDD の入力電圧で動作するように設計されています。VDD 超、または VSS 未満のコモンモード電圧での挙動は保証外です。

リーク電流は、OPA+ 入力 / 入力におけるソース / シンク電流です。リーク電流の影響を最小限に抑えるには、OPA+ 入力と OPA- 入力に接続されている実効インピーダンスをできるだけ小さく、かつ等しく保つ必要があります。

入力オフセット電圧は、線形領域で OPA を使う開ループ回路での OPA+ 入力と OPA- 入力の電圧差を示します。オフセット電圧は、入力オフセットに回路のゲインを掛けた DC オフセットとして出力に表れます。入力オフセット電圧は、コモンモード電圧の影響も受けません。OPA は、モジュールの入力オフセット電圧を最小限に抑えるよう工場での校正済みです。

開ループゲインは、差動入力電圧 (OPA+) - (OPA-) に対する出力電圧の比です。ゲインは DC 時に最大であり、周波数に応じて小さくなります。

ゲイン帯域幅積 (GBWP) は、開ループゲインが 0 dB となる周波数です。

29.2 OPA モジュールの制御

OPA モジュールを有効にするには、OPAxCON レジスタ ([レジスタ 29-1](#)) の OPAXEN ビットをセットします。OPA を有効にすると、OPAxOUT ピンの出力ドライバと OPA 出力の競合を防ぐために、この出力ドライバが強制的に 3 ステートになります。

Note: OPA モジュールを有効にすると、ポート デジタル ドライバではなくオペアンプ出力が OPAXOUT ピンを駆動します。オペアンプ出力駆動に関しては [表 36-17: オペアンプ \(OPA\)](#) を参照してください。

29.2.1 ユニティゲイン モード

OPAxCON レジスタ ([レジスタ 29-1](#)) の OPAXUG ビットはユニティゲイン モードを選択します。ユニティゲインを選択すると、OPA 出力が反転入力に接続され OPAXIN ピンを解放するため、このピンを汎用入出力に使えます。

29.2.2 プログラマブルな入力源選択

反転および非反転入力源は、それぞれ OPAXNCHS ([レジスタ 29-3](#)) および OPAXPCHS ([レジスタ 29-4](#)) レジスタを使って選択できます。入力源には以下のものが含まれます。

- 内蔵 DAC
- デバイスピンの
- 内蔵スロープ補償ランプ ジェネレータ
- デバイス内の他のオペアンプ

29.3 オーバーライド制御

29.3.1 オーバーライド モード

オペアンプは以下の 2 つの方法で動作をオーバーライドできます。

- 強制的 3 ステート出力
- 強制的ユニティゲイン

オーバーライド モードは OPxCON レジスタ ([レジスタ 29-1](#)) の ORM<1:0> ビットで選択します。オーバーライド モードは、モードが選択されかつオーバーライド信号が真の場合に有効になります。

29.3.2 オーバーライド信号源

オーバーライド信号源は OPAXORS レジスタ ([レジスタ 29-2](#)) で選択します。これらの信号源は以下の内蔵周辺モジュールからのものです。

- CCP 出力
- PWM 出力
- コンパレータ出力
- ゼロクロス検出出力
- 構成可能なロジックセル出力
- COG 出力

29.3.3 オーバーライド入力極性

オーバーライドの入力極性を反転する事で、選択した信号源が High レベルの時でも Low レベルの時でもオーバーライドを発生させる事ができます。オーバーライド極性は OPAXCON レジスタ ([レジスタ 29-1](#)) の ORPOL ビットで制御できます。

29.4 リセットの影響

デバイスリセットが発生すると、全てのレジスタはリセット状態に戻ります。これにより OPA モジュールは無効になります。

29.5 スリープの影響

オペアンプはスリープ中も動作を続けます。

PIC16(L)F1764/5/8/9

29.6 レジスタ定義：オペアンプ制御

表29-1にオペアンプ周辺モジュールの完全ビット名の接頭辞を示します。詳細は[セクション 1.1「レジスタおよびビット命名法」](#)を参照してください。

表 29-1:

周辺モジュール	ビット名の接頭辞
OPA1	OPA1
OPA2 ⁽¹⁾	OPA2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 29-1: OPAXCON: オペアンプ (OPAx) 制御レジスタ

R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	—	UG	—	ORPOL	ORM<1:0>	
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

- bit 7 **EN:** オペアンプ イネーブルビット
1 = オペアンプを有効にする
0 = オペアンプを無効にする (消費電力ゼロ)
- bit 6-5 **未実装:** 「0」として読み出し
- bit 4 **UG:** オペアンプ ユニティゲイン選択ビット
1 = OPA 出力を反転入力に接続する。OPAxIN- ピンは汎用 I/O として使える
0 = 反転入力を OPAXIN- ピンに接続する
- bit 3 **未実装:** 「0」として読み出し
- bit 2 **ORPOL:** オペアンプ オーバーライド入力極性ビット
1 = オーバーライド入力極性を反転する。オーバーライドは信号源が High の時に発生する
0 = オーバーライド入力極性を反転しない。オーバーライドは信号源が Low の時に発生する
- bit 1-0 **ORM<1:0>:** オペアンプ オーバーライド モード選択ビット
11 = 予約済み使用不可
10 = オーバーライド信号が真の場合、オペアンプを強制的にユニティゲインに設定する
01 = オーバーライド信号が真の場合、オペアンプ出力を 3 ステートに設定する
00 = 出力オーバーライド機能を無効にする

レジスタ 29-2: OPAxORS: オペアンプ オーバーライド信号源選択ビット

U-0	U-0	U-0	R/W-0/0	R/W-0/x	R/W-0/x	R/W-0/0	R/W-0/x	
—	—	—	ORS<4:0>					
bit 7								bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = 条件により異なる

bit 7-5 **未実装:** 「0」として読み出し

bit 4-0 **ORS<4:0>:** オペアンプ出力オーバーライド信号源選択ビット

11111 = 予約済み使用不可
 .
 .
 .
 10110 = 予約済み使用不可
 10101 = COG2D⁽¹⁾ をオーバーライド信号源にする
 10100 = COG2C⁽¹⁾ をオーバーライド信号源にする
 10011 = COG2B⁽¹⁾ をオーバーライド信号源にする
 10010 = COG2A⁽¹⁾ をオーバーライド信号源にする
 10001 = COG1C をオーバーライド信号源にする
 10000 = COG1C をオーバーライド信号源にする
 01111 = COG1B をオーバーライド信号源にする
 01110 = COG1A をオーバーライド信号源にする
 01101 = LC3_out をオーバーライド信号源にする
 01100 = LC2_out をオーバーライド信号源にする
 01011 = LC1_out をオーバーライド信号源にする
 01010 = ZCD1_output をオーバーライド信号源にする
 01001 = sync_C4OUT⁽¹⁾ をオーバーライド信号源にする
 01000 = sync_C3OUT⁽¹⁾ をオーバーライド信号源にする
 00111 = sync_C2OUT をオーバーライド信号源にする
 00110 = sync_C1OUT をオーバーライド信号源にする
 00101 = PWM6_out⁽¹⁾ をオーバーライド信号源にする
 00100 = PWM5_out をオーバーライド信号源にする
 00011 = PWM4_out⁽¹⁾ をオーバーライド信号源にする
 00010 = PWM3_out をオーバーライド信号源にする
 00001 = CCP2_out⁽¹⁾ をオーバーライド信号源にする
 00000 = CCP1_out をオーバーライド信号源にする

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 29-3: OPAXNCHS: オペアンプ負チャンネル信号源選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCH<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-4 **未実装:** 「0」として読み出し

bit 3-0 **NCH<3:0>:** オペアンプ反転入力チャンネル選択ビット

1111 = 予約済み使用不可

⋮

1010 = 予約済み使用不可

1001 = プログラマブル ランプ ジェネレータ PRG2_out⁽¹⁾

1000 = プログラマブル ランプ ジェネレータ PRG1_out

0111 = 予約済み使用不可

0110 = FVR_Buffer2

0101 = DAC4_out⁽¹⁾

0100 = DAC3_out

0011 = DAC2_out⁽¹⁾

0010 = DAC1_out

0001 = OPAXIN1- ピン⁽¹⁾

0000 = OPAXIN0- ピン

Note 1: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

レジスタ 29-4: OPAxPCHS: オペアンプ正チャンネル信号源選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	PCH<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = 条件により異なる

bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **PCH<3:0>:** オペアンプ非反転入力チャンネル選択ビット
1111 = 予約済み使用不可
. . .
1010 = 予約済み使用不可
1001 = プログラマブル ランプ ジェネレータ PRG2_out⁽¹⁾
1000 = プログラマブル ランプ ジェネレータ PRG1_out
0111 = 予約済み使用不可
0110 = FVR_Buffer2
0101 = DAC4_out⁽¹⁾
0100 = DAC3_out
0011 = DAC2_out⁽¹⁾
0010 = DAC1_out
0001 = OPAxIN1+ ピン⁽¹⁾
0000 = OPAxIN0+ ピン

Note 1: PIC16(L)F1768/9 のみです。

表 29-2: オペアンプ関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELB ⁽²⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC ⁽²⁾	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
ANSELC ⁽³⁾	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
DACxCON0	EN	FM	OE1	—	PSS<1:0>		NSS<1:0>		193
DACxREF	—	—	—	REF<4:0>					193
DACxREFL ⁽²⁾	REF<7:0>								200
DACxREFH ⁽²⁾	REF<15:8>								200
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173
OPAxCAN	EN	—	—	UG	—	ORPOL	ORM<1:0>		348
OPAxNCHS	—	—	—	—	NCH<3:0>				350
OPAxPCHS	—	—	—	—	PCH<3:0>				351
OPAxORS	—	—	—	ORS<4:0>					349
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC ⁽²⁾	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TRISC ⁽³⁾	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	1	0	150

凡例: — = 未実装、「0」として読み出し。網掛けの部分はオペアンプでは使いません。

- Note 1:** 未実装、「1」として読み出します。
2: PIC16(L)F1768/9 のみです。
3: PIC16(L)F1764/5 のみです。

30.0 プログラマブル ランプ ジェネレータ (PRG) モジュール

プログラマブル ランプ ジェネレータ (PRG) は、立ち上がりおよび立ち下がり直線ランプを生成するためのモジュールです。代表的な応用回路であるスロープ補償は、固定周波数、連続電流、電流モードの各種スイッチング電源に使われます。スロープ補償は 50% 以上のデューティ サイクルにおける周波数の不安定性を回避できるため、これらの電源に必要な機能です。

PRG には以下の機能があります。

- 正負電圧直線ランプ出力
- プログラマブルな電流ソース / シンク
- 内部 / 外部参照電圧を選択可能
- 内部 / 外部タイミング源を選択可能

図 30-1 に PRG の概略ブロック図を示します。

30.1 基本動作

PRG は以下に示す 3 つの電圧ランプ ジェネレータモードで動作できます。

- 立ち下がり電圧 (スロープ補償)
- 立ち上がり電圧
- 交互立ち上がり立ち下がり電圧

立ち上がりモードまたは立ち下がりモードでは、内蔵コンデンサは `set_falling` タイミング入力が入真になると放電され、`set_rising` タイミング入力が入真になると内部生成した定電流で充電されます。その結果、直線ランプ波形は設定した電圧入力レベルからスタートし、`set_falling` タイミング入力が入真でランプ波形が終了して元のレベルにリセットされます。両タイミング入力が入真になるかは `set_falling` 入力により決まります。

シングルエンドの信号源でこの動作を制御するには、`set_rising` および `set_falling` 入力の両方に同じ信号源を選択し、片方の極性を対応する極性制御ビットで反転します。

交互モードでは、コンデンサは放電されず 1 方向に充電され次に反対方向に充電される事を繰り返します。

入力選択は全てのモードで同じです。入力電圧は下記のいずれかから供給します。

- PRGxIN0 または PRGxIN1 ピン
- 内部の固定参照電圧 (FVR) のバッファ出力
- 内蔵 DAC

タイミング入力源は下記から選択します。

- コンパレータの同期出力
- PWM 出力
- I/O ピン

ランプ出力は全てのコンパレータまたはオペアンプの入力として使うことができます。

30.1.1 スロープ補償

スロープ補償は、各 PWM 周期の開始時に内蔵コンデンサを瞬時に放電させる事で機能します。内蔵コンデンサの片方の電極は電圧入力源に接続され、もう片方は内蔵電流シンクに接続されています。内蔵電流シンクはこのコンデンサをプログラマブルな速度で充電します。コンデンサが充電するに従ってコンデンサの電極間の電圧は電圧源から差し引かれていき、目的とする速度の直線的な電圧の低下が得られます (図 30-2 参照)。 `set_falling` タイミング入力が入真になると、コンデンサが放電しランプは終了します。次のランプは、`set_rising` タイミング入力が入真になった時に開始します。PRGxCON0 レジスタの OS ビットをセットしてオプションのワンショットを有効にすると、少なくともワンショット周期 (50 ns typ.) の間 `set_rising` タイミング入力に優先して短絡スイッチを閉じておく事で、コンデンサを完全に放電できます。ワンショット周期の間のエッジセンス タイミング入力は無視されます。ワンショット周期の間に発生し、ワンショット周期を超えて継続するレベルセンス タイミング入力はワンショット期間の終わりまで保留されます。

30.1.2 ランプ生成

ランプ生成はスロープ補償と似ていますが、ランプ生成ではスロープが立ち上がり立ち下がりの両方立ち上がりだけのどちらかです。

30.1.2.1 交互立ち上がり / 立ち下がりランプ

交互立ち上がり / 立ち下がりランプ生成機能は、内蔵電流源と電流シンクを使い、内蔵アナログスイッチとタイミング入力源の同期制御を利用して、モジュール出力電圧をランプアップしその後ランプダウンする事で機能します。

初期化後、出力電圧は電流源によってプログラマブルな速度で `set_falling` タイミング入力が入真になるまで直線的に上昇します。`set_falling` タイミング入力が入真になった時電流源が切り離されます。それと同時に電流シンクが接続され、同様にプログラマブルな速度で出力電圧を `set_rising` タイミング入力が入真になるまで直線的に下降させ、これによりランプスロープが反転します。この過程を繰り返す事で、のこぎり波形を生成します (図 30-3、図 30-4 参照)。

`set_rising` および `set_falling` タイミング入力は、PRGxCON0 レジスタの REDG および FEDG ビットで、それぞれエッジセンスまたはレベルセンスのどちらにも設定できます。クロックのような周期信号にはエッジセンス動作を、コンパレータ出力等のアナログリミットリガにはレベルセンス動作を推奨します。

ワンショットが有効な場合 (OS ビットはセット)、立ち下がりおよび立ち上がりランプは少なくともワンショット周期の間持続します。ワンショット周期の間のエッジセンス タイミング入力は無視されます。ワンショット周期の間に発生し、ワンショット周期を超えて継続するレベルセンス タイミング入力はワンショット期間の終わりまで保留されます。

30.1.2.2 立ち上がりランプ

立ち上がりランプモードは、ランプに立ち上がりスロープがあるだけで立ち下がりスロープがない点だけがスロープ補償モードと異なります。内蔵コンデンサの片方の電極は電圧入力源に接続され、もう片方は内蔵電流源に接続されています。内蔵電流源はこのコンデンサをプログラブルな速度で充電します。コンデンサが充電するに従ってコンデンサの電極間の電圧が電圧源に加算されていき、目的とする速度の直線的な電圧の立ち上がり得られます(図30-5参照)。set_falling タイミング入力が入力が真になると、コンデンサが放電しランプは終了します。次のランプは、set_rising タイミング入力が入力が真になった時に開始します。

PRGxCON0 レジスタの OS ビットをセットしてオプションのワンショットを有効にすると、少なくともワンショット周期 (50 ns typ.) の間 set_rising タイミング入力に優先して短絡スイッチを閉じておく事で、コンデンサを完全に放電できます。ワンショット周期の間のエッジセンス タイミング入力は無視されます。ワンショット周期の間に発生し、ワンショット周期を超えて継続するレベルセンス タイミング入力はワンショット期間の終わりまで保留されます。

30.2 EN、RDY、GO

PRGxCON0 レジスタの EN ビットで電流源を含むアナログ回路を有効にします。これによって PRG モジュールを使う準備ができ、モジュールを動作状態にする前に安定化させる事ができます。EN ビットをセットするとタイミング入力が有効になり、GO ビットがセットされる前に最初のランプ動作を決定できます。EN ビットをセットするとコンデンサ短絡スイッチが閉じ、GO ビットが「0」の間閉じたままです。

PRGxCON1 レジスタの RDY ビットはアナログ回路および電流源が安定している事を示します。

PRGxCON0 レジスタの GO ビットでスイッチ制御回路を有効にし、PRG を動作状態にします。GO がクリアからセットに遷移するとワンショットをトリガし、その結果コンデンサ短絡スイッチを閉じたままワンショット周期の間維持します。

動作を確実にするため、最初に EN ビットをセットした後 RDY ビットが High になるのを待ってから GO ビットをセットします。

30.3 独立した Set_rising および Set_falling タイミング入力

タイミング入力でランプの開始および終了時期を決定します。交互立ち上がり / 立ち下がりモードでは set_rising 入力が入力が真になった時ランプが立ち上がり、set_falling 入力が入力が真になった時立ち下がります。スロープ補償モードと立ち上がりランプモードでは、set_falling タイミング入力が入力が真になった時コンデンサが放電され、set_rising タイミング入力が入力が真になった時ランプが開始します。set_falling 入力が入力が真になった時左右します。

30.4 レベルセンスとエッジセンスのタイミング

set_rising および set_falling タイミング入力は、レベルセンスまたはエッジセンスとして独立に設定できます。

オーバーライド条件が終わった後タイミング入力が真である状態を検出する必要がある場合に、レベルセンス動作は便利です。例えば、ワンショットのオーバーライド動作で抑制された可能性があるコンデンサ生成タイミング入力に対してレベルセンスは便利です。レベルセンスを使うと、ワンショット周期内で変化するコンデンサの出力をワンショット期間の末尾で検出できます。エッジセンスではこの変化は無視されます。

エッジセンス動作は、PWM、クロック等で生成された周期的なタイミング入力に対して便利です。レベルセンスの場合、周期信号のデューティ サイクルによっては他のタイミング入力と干渉する場合があります。set_rising タイミング入力源としてレベルセンスの 50% PWM、set_falling タイミング入力源としてレベルセンスのコンパレータをそれぞれ備える交互ランプモードを例として取り上げます。PWM 信号がまだ High のうちにコンパレータ出力がランプを反転させると、コンパレータ信号が Low になった時ランプは再び反転してしまいます。エッジセンスの set_rising タイミング入力ではこれと同じシナリオでもランプ出力は正しく変化し、PWM 信号の立ち上がりエッジだけで立ち上がります。

エッジセンスの set_rising および set_falling タイミング入力は、PRGxCON1 レジスタの REDG および FEDG ビットでそれぞれ設定できます。

30.5 ワンショット最小タイミング

ワンショット タイマは、スロープ補償および立ち上がりランプモードでは最小のコンデンサ放電時間を保証し、交互ランプモードでは最小の立ち上がりまたは立ち下がりランプ継続時間を保証します。PRGxCON0 レジスタの OS ビットをセットするとワンショットタイマが有効になります。

30.6 DAC 電圧源

電圧源として DAC を使うと、電流と DAC 等価抵抗の積に等しい電圧オフセットが生じます。これはスロープ補償およびランプモードでは一定のオフセットとなり、交互モードでは正 / 負のステップ オフセットとなります。この問題を避けるには、ユニティ ゲインに設定したオペアンプを通して DAC 出力を PRG 入力に接続します。

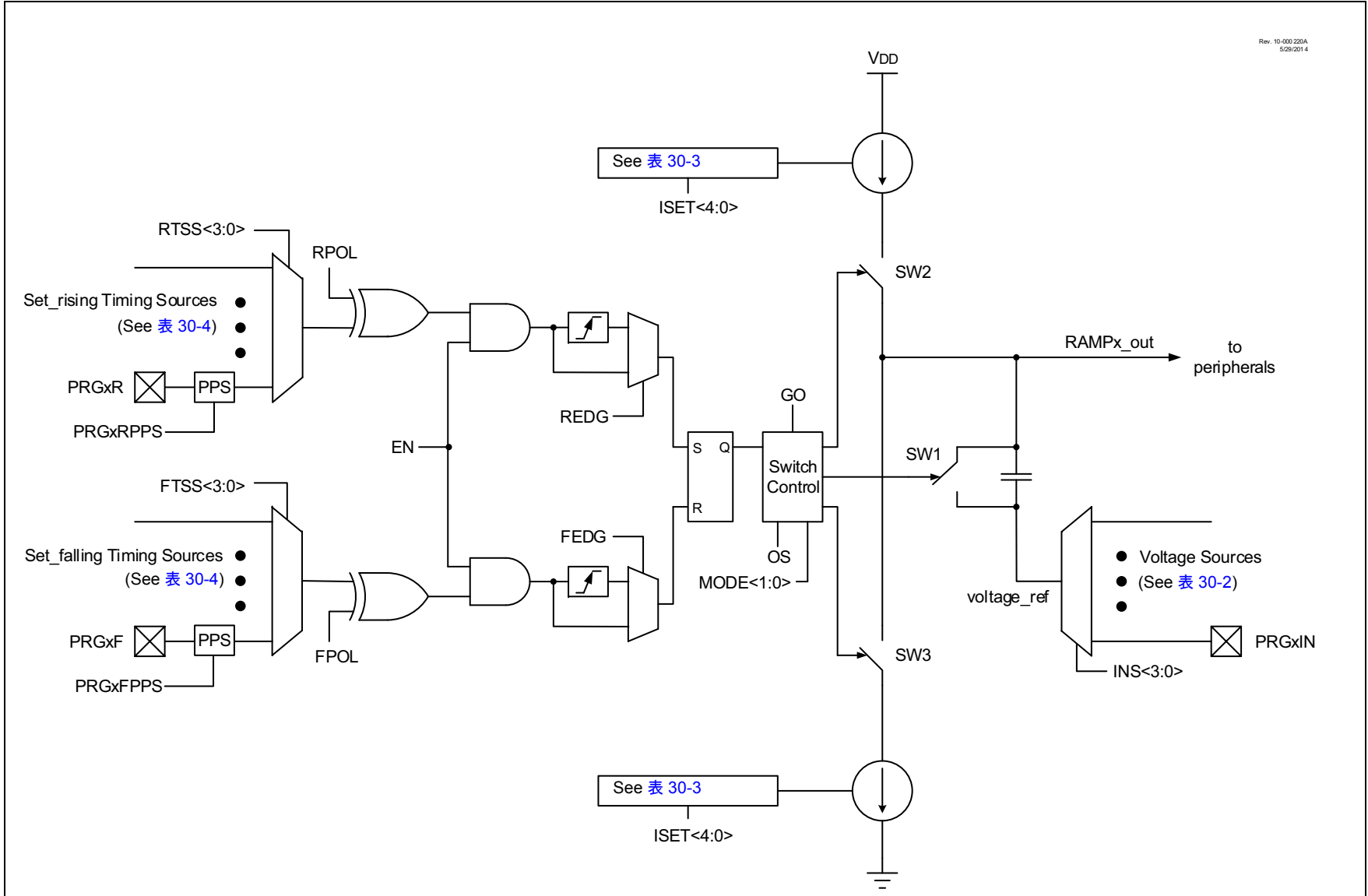
30.7 スリープ中の動作

RG モジュールはスリープで影響を受けません。

30.8 リセットの影響

RG モジュールは無効状態にリセットします。

図 30-1: PRG の概略ブロック図



Rev. 10-00 223A
5/2/2014

図 30-2: スロープ補償 (立ち下がりがランプ) のタイミング図 (MODE = 00)

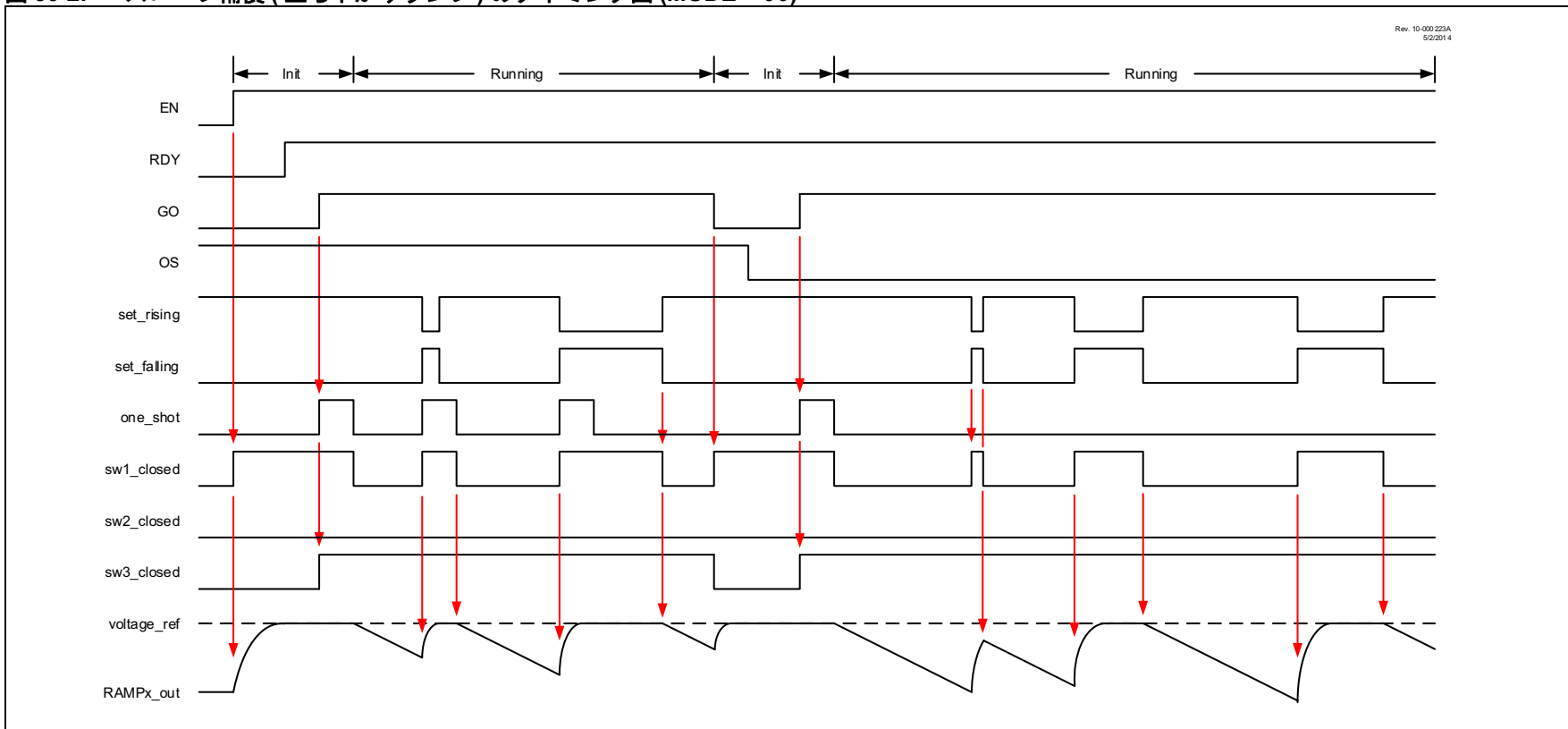
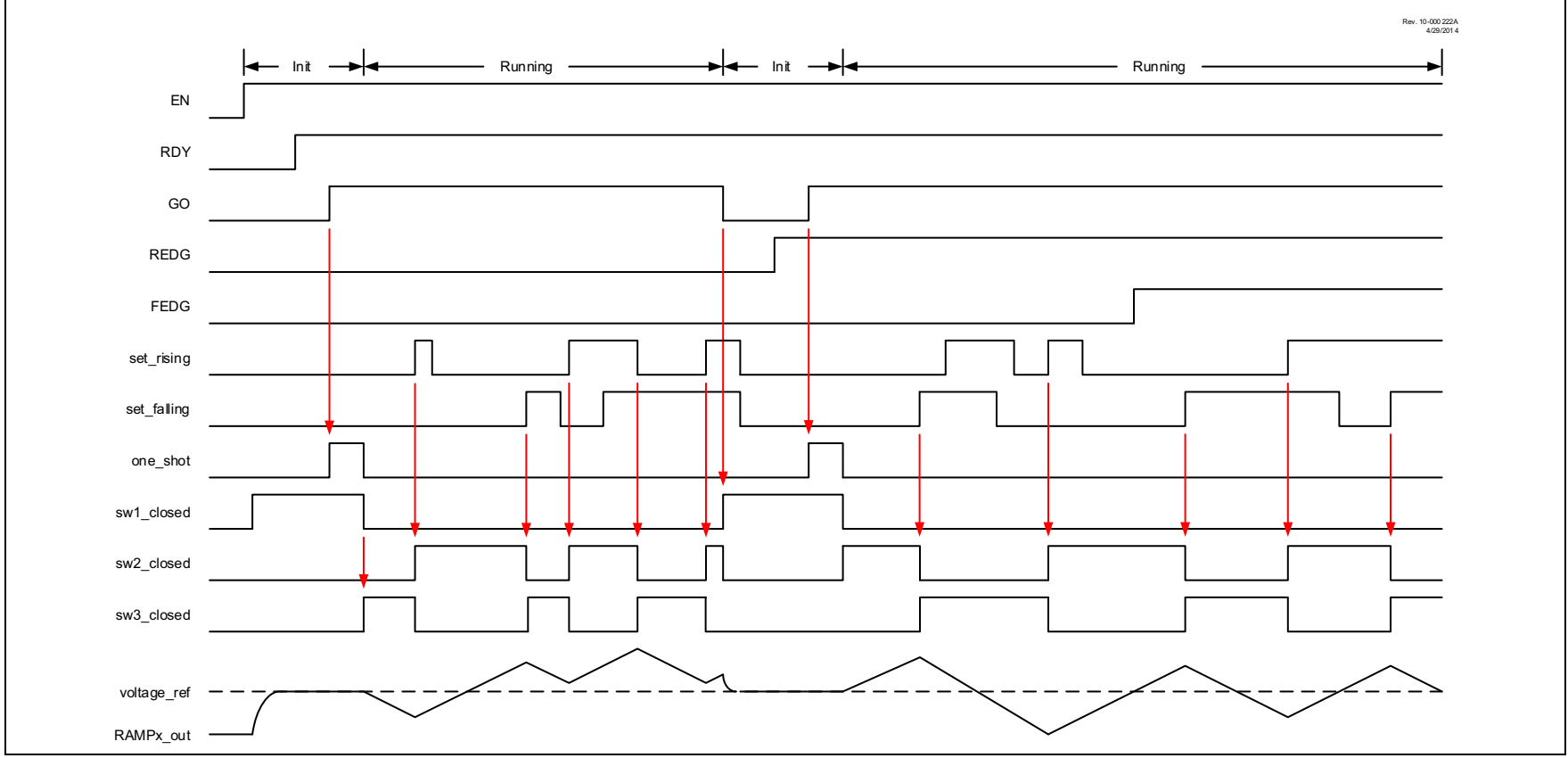


図 30-3: 交互立ち上がり / 立ち下がりランプ生成のタイミング図 (OS = 0、MODE = 01)

Rev. 10-00 222A
4/29/2014



Rev. 10-000 225A
6/2/2014

図 30-4: 交互立ち上がり / 立ち下がりランプ生成のタイミング図 (OS = 1、MODE = 01)

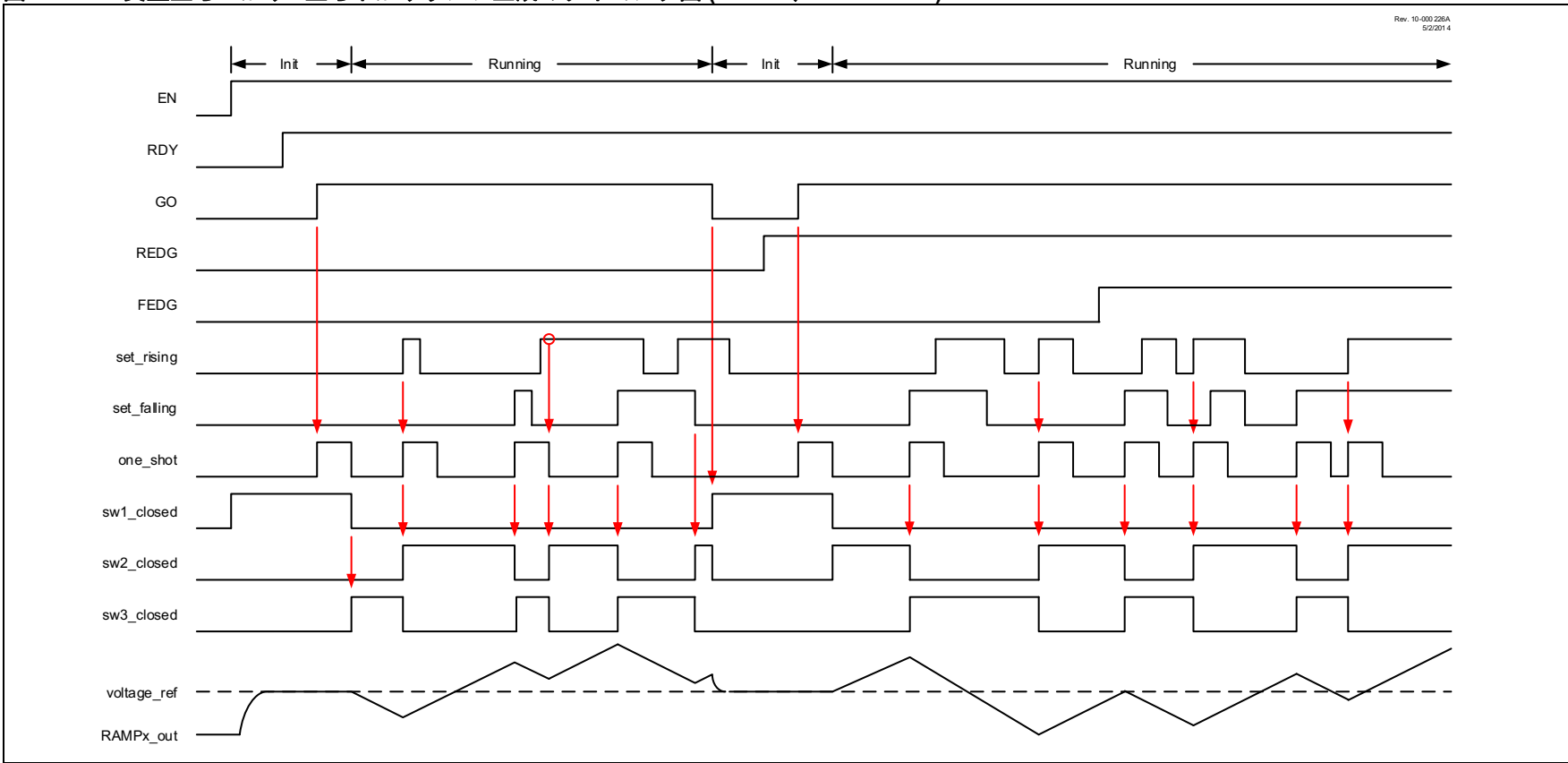
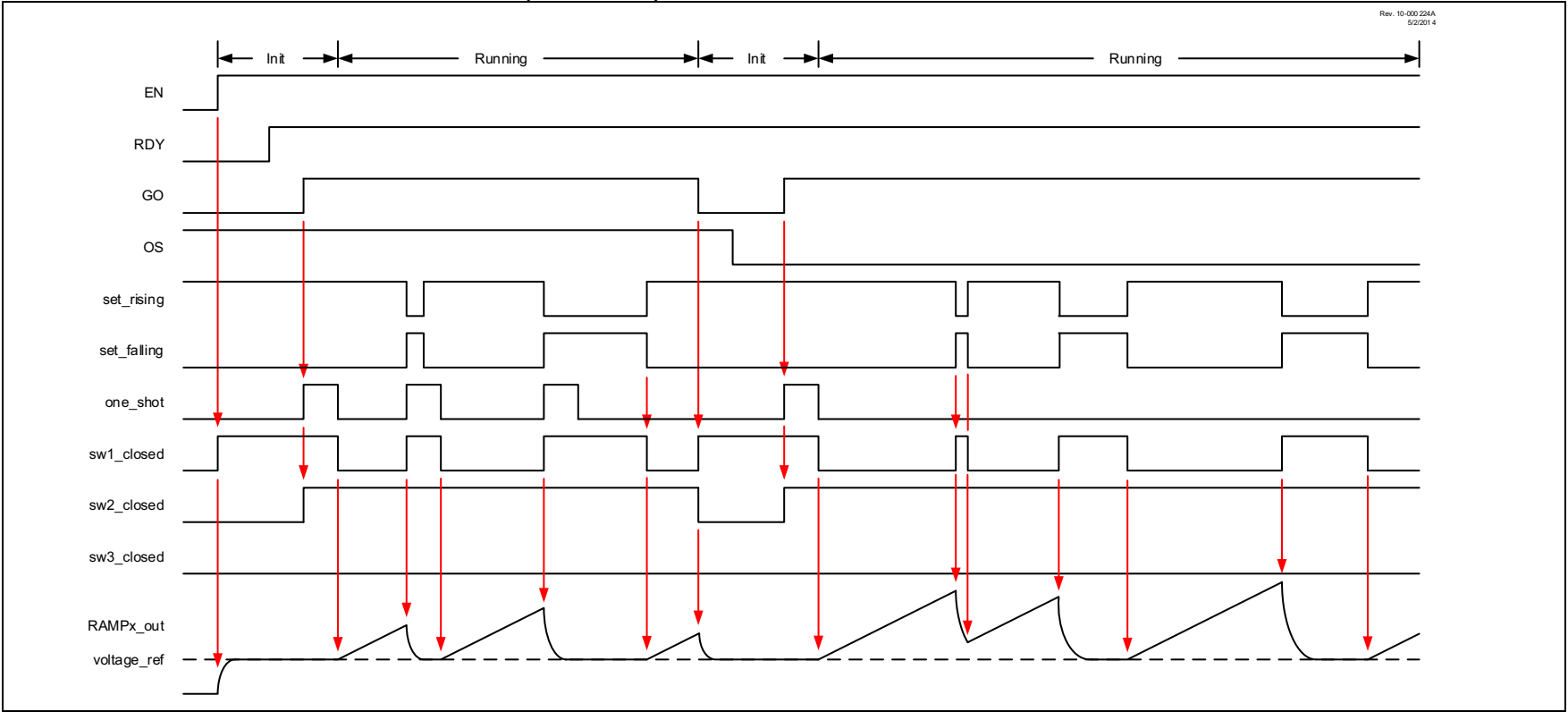


図 30-5: 立ち上がりランプ生成のタイミング図 (MODE = 10)



30.9 スロープ補償の応用回路

スロープ補償の回路例を図 30-6 に示します。PRG 入力電圧は PRGxIN であり、I/O ピンをオペアンプ出力と共有しています。オペアンプ出力はピーク電流検出電圧の期待値 (VREF) で動作するためのものです。PRG 出力電圧は VREF から開始し、目標の回路電流検出電圧の立ち上がり速度の半分以下の速度で低下します。従って、補償回路のスロープ (V/μs) は式 30-1 で計算できます。

式 30-1:

$$\frac{V}{\mu s} \geq \frac{\frac{V_{REF}}{2}}{PWM \text{ Period } (\mu s)}$$

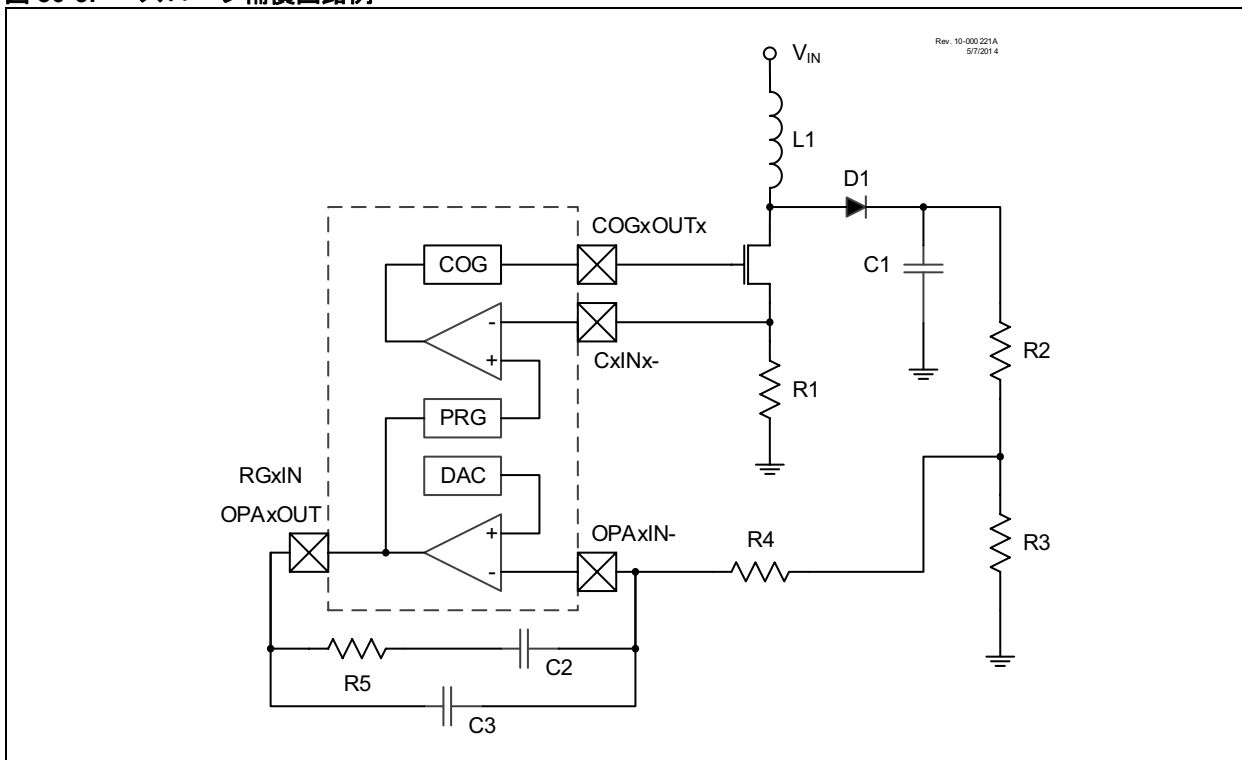
例えば、回路で 1 Ω の電流検出抵抗を使いピーク電流 1 A が流れた時、ピーク電流による電圧降下は 1 V となります。従ってこの時にオペアンプ出力が動作するように設計します。電源の PWM 周波数が 1 MHz の場合、周期は 1 μs です。従って、必要なスロープは式 30-2 から、0.5 V/μs と求められます。

式 30-2:

$$\frac{\frac{V_{REF}}{2}}{PWM \text{ Period } (\mu s)} = \frac{\frac{1}{2}}{1 \mu s} = 0.5 V / \mu s$$

Note: 0.5 V/μs の設定は ISET<4:0> = 6 です。

図 30-6: スロープ補償回路例



PIC16(L)F1764/5/8/9

30.10 レジスタ定義：スロープ補償制御

表 30-1 に PRG 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1 「レジスタおよびビット命名法」](#) を参照してください。

表 30-1:

周辺モジュール	ビット名の接頭辞
PRG1	RG1
PRG2 ⁽¹⁾	RG2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 30-1: PRGxCON0: プログラマブル ランプ ジェネレータ制御 0 レジスタ

R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	FEDG	REDG	MODE<1:0>		OS	GO
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

- bit 7 **EN:** プログラマブル ランプ ジェネレータ イネーブルビット
1 = PRG モジュールを有効にする
0 = PRG モジュールを無効にする
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **FEDG:** Set_falling 入力モード選択ビット
1 = Set_falling タイミング入力をレベルセンスにする
0 = Set_falling タイミング入力をエッジセンスにする
- bit 4 **REDG:** Set_rising 入力モード選択ビット
1 = Set_rising タイミング入力をレベルセンスにする
0 = Set_rising タイミング入力をエッジセンスにする
- bit 3-2 **MODE<1:0>:** プログラマブル ランプ ジェネレータ モード選択ビット
11 = 予約済み
10 = 立ち上がりランプ ジェネレータ
01 = 交互立ち上がり / 立ち下がりランプ ジェネレータ
00 = スロープ補償
- bit 1 **OS:** ワンショット イネーブルビット
1 = ワンショットを有効にする。内部的にワンショットでタイミング制御される事で、コンデンサは最小限の時間で放電される
0 = ワンショットを無効にする。コンデンサはタイミング入力が高レベルの間放電される
- bit 0 **GO:** ランプ生成制御開始ビット
EN = 0 の場合
このビットは常に「0」に固定される
EN = 1 の場合
1 = スロープまたはランプ機能を動作させる
0 = スロープまたはランプ機能を動作させない。全ての電流源スイッチは開き、コンデンサ放電スイッチは閉じる

PIC16(L)F1764/5/8/9

レジスタ 30-2: PRGxCON1: プログラマブル ランプ ジェネレータ制御 1 レジスタ

U-0	U-0	U-0	U-0	U-0	R-0	R/W-0/0	R/W-0/0
—	—	—	—	—	RDY	FPOL	RPOL
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

- bit 7-3 **未実装:** 「0」として読み出し
- bit 2 **RDY:** スロープ ジェネレータ レディ ステータスビット
1 = PRG の準備が完了している
0 = PRG の準備は完了していない
- bit 1 **FPOL:** 立ち下がりがイベント極性選択ビット
1 = Set_falling タイミング入力をアクティブ Low にする
0 = Set_falling タイミング入力をアクティブ High にする
- bit 0 **RPOL:** 立ち上がりがイベント極性選択ビット
1 = Set_rising タイミング入力をアクティブ Low にする
0 = Set_rising タイミング入力をアクティブ High にする

レジスタ 30-3: PRGxINS: 電圧入力選択レジスタ

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	INS<2:0>		
bit 7						bit 0	

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

- bit 7-3 **未実装:** 「0」として読み出し
- bit 2-0 **INS<2:0>:** 電圧入力選択ビット
ランプが始まる電圧レベルの入力源を選択します。表 30-2 を参照してください。

表 30-2: 電圧入力源

INS<2:0>	PIC16(L)F1764/5 の電圧源	PIC16(L)F1768/9 の電圧源	
111	予約済み	予約済み	予約済み
110	予約済み	DAC4_output	DAC4_output
101	DAC3_output	DAC3_output	DAC3_output
100	予約済み	DAC2_output	DAC2_output
011	DAC1_output	DAC1_output	DAC1_output
010	FVR_buffer1	FVR_buffer1	FVR_buffer1
001	予約済み	PRG1IN1/OPA2OUT	PRG2IN1/OPA1OUT
000	PRG1IN0/OPA1OUT	PRG1IN0/OPA1OUT	PRG2IN0/OPA2OUT

PIC16(L)F1764/5/8/9

レジスタ 30-4: PRGxCON2: プログラマブル ランプ ジェネレータ制御 2 レジスタ

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ISET<4:0>				
bit 7							bit 0

凡例:		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	q = コンフィグレーション ビット依存

bit 7-5 **未実装:** 「0」として読み出し

bit 4-0 **ISET<4:0>:** PRG 電流ソース / シンク設定ビット
電流ソース / シンクとスロープ速度を設定します。表 30-3 を参照してください。

表 30-3: プログラマブル ランプ ジェネレータ電流設定

ISET<4:0>	電流設定 (uA)	スロープ速度 (V/us)	ISET<4:0>	電流設定 (uA)	スロープ速度 (V/us)
0h	2	0.2	10h	10	1.0
1h	2.5	0.25	11h	11	1.1
2h	3	0.3	12h	12	1.2
3h	3.5	0.35	13h	13	1.3
4h	4	0.4	14h	14	1.4
5h	4.5	0.45	15h	15	1.5
6h	5	0.5	16h	16	1.6
7h	5.5	0.55	17h	17	1.7
8h	6	0.6	18h	18	1.8
9h	6.5	0.65	19h	19	1.9
Ah	7	0.7	1Ah	20	2.0
Bh	7.5	0.75	1Bh	21	2.1
Ch	8	0.8	1Ch	22	2.2
Dh	8.5	0.85	1Dh	23	2.3
Eh	9	0.9	1Eh	24	2.4
Fh	9.5	0.95	1Fh	25	2.5

レジスタ 30-5: PRGxRTSS: Set_rising タイミング源選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	RTSS<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = コンフィグレーション ビット依存

bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **RTSS<3:0>:** Set_rising タイミング源選択ビット
表 30-4 を参照してください。

レジスタ 30-6: PRGxFTSS: Set_falling タイミング源選択レジスタ

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	FTSS<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア q = コンフィグレーション ビット依存

bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **FTSS<3:0>:** Set_falling タイミング源選択ビット
表 30-4 を参照してください。

表 30-4: プログラマブル ランプ ジェネレータ タイミング源

RTSS<3:0>/FTSS<3:0>	タイミング源	RTSS<3:0>/FTSS<3:0>	タイミング源
0000	sync_C1OUT	1000	PWM6_output ⁽²⁾
0001	sync_C2OUT	1001	CCP1_out
0010	sync_C3OUT ⁽²⁾	1010	CCP2_out ⁽²⁾
0011	sync_C4OUT ⁽²⁾	1011	予約済み
0100	PRGxR/PRGxF Pin ⁽¹⁾	1100	予約済み
0101	PWM3_output	1101	予約済み
0110	PWM4_output ⁽²⁾	1110	予約済み
0111	PWM5_output	1111	予約済み

Note 1: 入力ピンは PRGxRPPS レジスタまたは PRGxFPPS レジスタで選択します。
2: PIC16(L)F1768/9 のみです。

PIC16(L)F1764/5/8/9

表 30-5: PRG モジュール関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
PRG1CON0	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	360
PRG1CON1	—	—	—	—	—	RDY	FPOL	RPOL	361
PRG1CON2	—	—	—	ISET<4:0>					362
PRG1INS	—	—	—	—	—	INS<2:0>			361
PRG1RPPS	—	—	—	PRG1RPPS<4:0>					363
PRG1FPPS	—	—	—	PRG1FPPS<4:0>					363
PRG1RTSS	—	—	—	—	RTSS<3:0>				158、160
PRG1FTSS	—	—	—	—	FTSS<3:0>				158、160
PRG2CON0 ⁽¹⁾	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	360
PRG2CON1 ⁽¹⁾	—	—	—	—	—	RDY	FPOL	RPOL	361
PRG2CON2 ⁽¹⁾	—	—	—	ISET<4:0>					362
PRG2INS ⁽¹⁾	—	—	—	—	—	INS<2:0>			361
PRG2RPPS ⁽¹⁾	—	—	—	PRG2RPPS<4:0>					363
PRG2FPPS ⁽¹⁾	—	—	—	PRG2FPPS<4:0>					363
PRG2RTSS ⁽¹⁾	—	—	—	—	RTSS<3:0>				158、160
PRG2FTSS ⁽¹⁾	—	—	—	—	FTSS<3:0>				158、160
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	150
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
ANSEL	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	152

凡例: — = 未実装、「0」として読み出し。網掛けの部分は PRG モジュールでは使いません。

Note 1: PIC16(L)F1768/9 のみです。

31.0 データ信号モジュレータ (DSM)

データ信号モジュレータ (DSM) は、データストリーム (モジュレータ信号) と搬送波信号を混合して変調出力を生成できる周辺モジュールです。

DSM モジュールに供給する搬送波信号とモジュレータ信号には、デバイス内部の周辺モジュールの出力を使う事も、入力ピンからの外部信号を使う事もできます。

変調出力信号は、搬送波信号とモジュレータ信号の論理積 (AND) 演算を実行して生成され、MDxOUT ピンから出力されます。

この搬送波信号は2つの独立した信号から構成されます。搬送波 High 信号 (CARH) と搬送波 Low 信号 (CARL) です。モジュレータ信号 (MOD) が論理 High 状態にある間、DSM は搬送波 High 信号をモジュレータ信号と混合します。モジュレータ信号 (MOD) が論理 Low 状態にある間、DSM は搬送波 Low 信号をモジュレータ信号と混合します。

このような方法により、DSM モジュールは以下の変調方式で信号を生成できます。

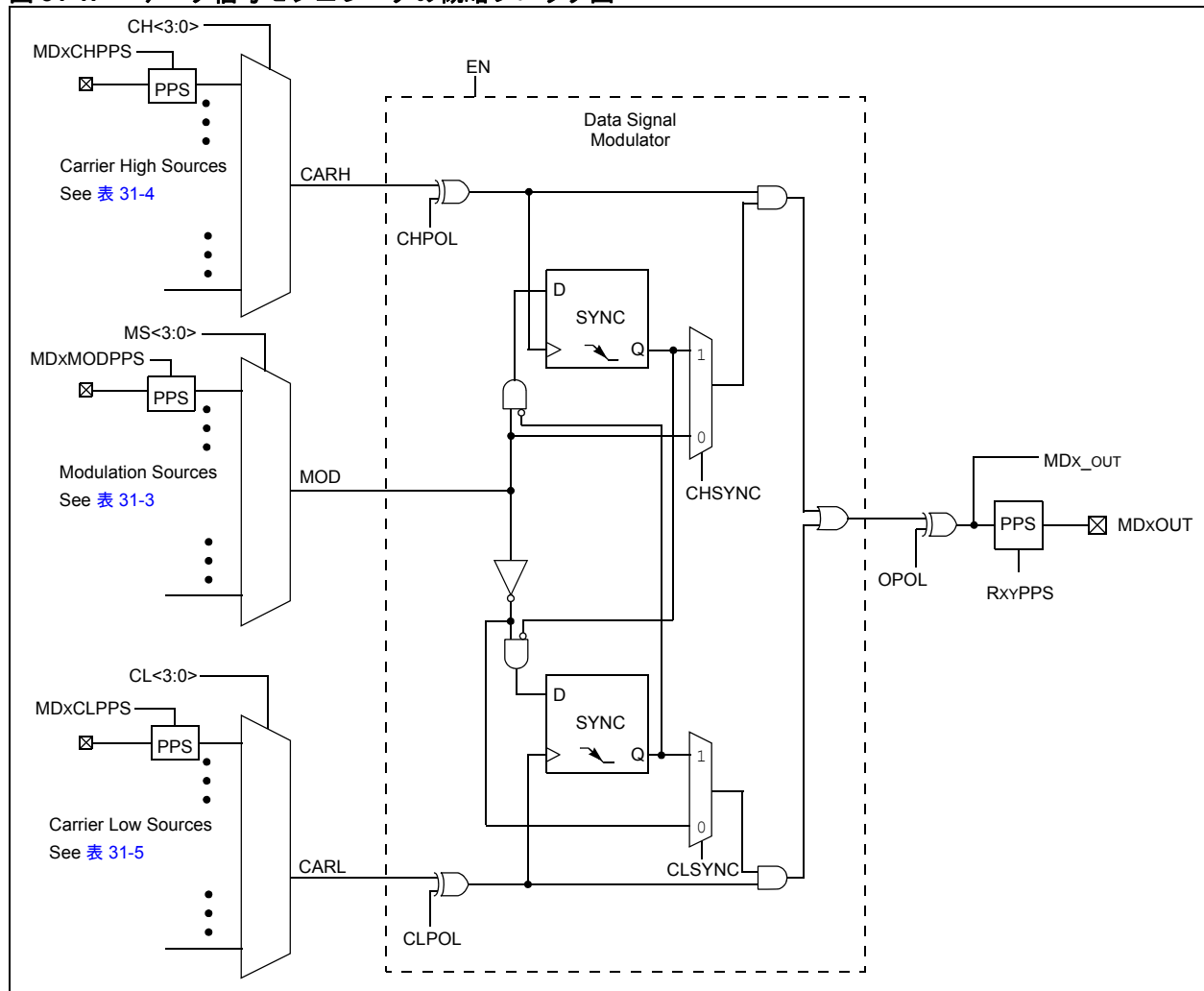
- 周波数シフトキーイング (FSK) 変調
- 位相シフトキーイング (PSK) 変調
- ON/OFF キーイング (OOK) 変調

さらに DSM モジュールは、以下の機能を提供します。

- 搬送波同期
- 搬送波信号源の極性選択
- 搬送波信号源のピンの無効化
- プログラマブルなモジュレータ データ
- モジュレータ信号源のピンの無効化
- モジュレータ出力の極性選択
- スルーレート制御

図31-1にDSMモジュールの概略ブロック図を示します。

図 31-1: データ信号モジュレータの概略ブロック図



PIC16(L)F1764/5/8/9

31.1 DSM の動作

DSM モジュールを有効にするには、MDxCON レジスタの EN ビットをセットします。MDxCON レジスタの EN ビットをクリアすると、搬送波 High および搬送波 Low 信号源が自動的に Vss に切り換わり、DSM モジュールは無効になります。モジュレータ信号源も MDxCON0 レジスタの BIT ビットに切り換わります。これによって DSM モジュールを確実に無効化するだけでなく、消費電流も最小化します。

変調源、変調 High 搬送波、変調 Low 搬送波の各制御レジスタに保持しているモジュレータ、搬送波 High、搬送波 Low の各信号源に対する設定値は、EN ビットをクリアして DSM モジュールを無効にしても影響を受けません。これらのレジスタ内の値は DSM が非アクティブの間も変化しません。搬送波 High、搬送波 Low、モジュレータの各信号源は、EN ビットをセットして DSM モジュールが有効にされ、アクティブになった時点で再度選択されます。

ピンの PPS 制御レジスタ (レジスタ 12-2 参照) で DSM モジュールを選択する事で、変調出力信号はどのデバイス I/O ピンにも出力できます。出力がどの I/O ピンにも指定されていない場合、DSM モジュールはアクティブ状態を維持し信号の混合を続けますが、その出力はどのピンにも転送されません。

31.2 モジュレータ信号源

モジュレータ信号は MDxSRC レジスタの MS<4:0> ビットで選択します。表 31-3 に選択方法を示します。

31.3 搬送波信号源

搬送波 High 信号源は MDxCARH レジスタの CH<3:0> ビットで選択します。表 31-4 に選択方法を示します。搬送波 Low 信号は MDxCARL レジスタの CL<3:0> ビットで選択します。表 31-5 に選択方法を示します。

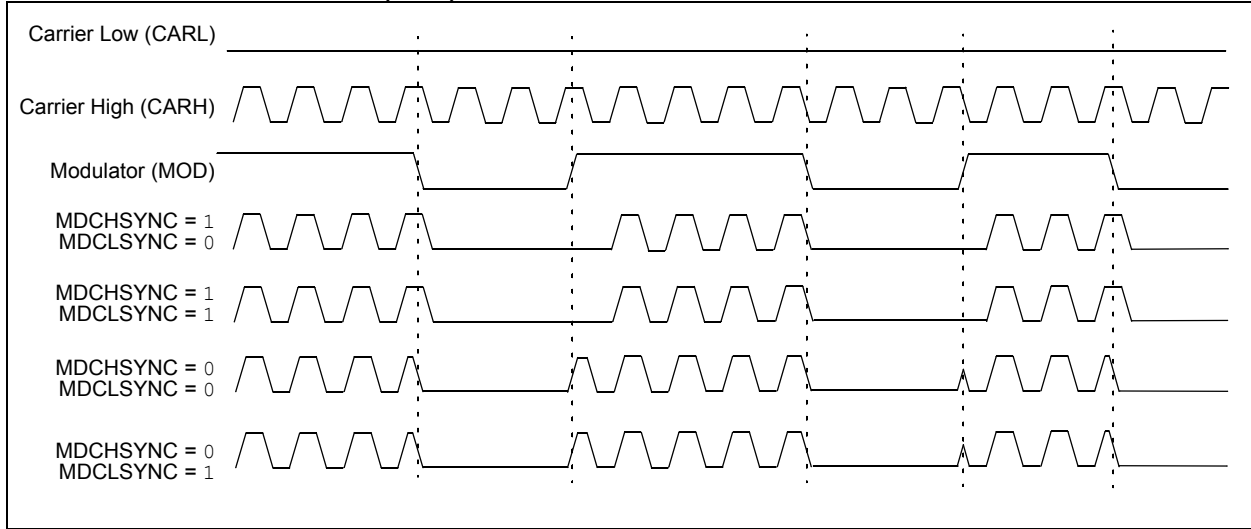
31.4 搬送波同期

DSM が搬送波信号源を搬送波 High と搬送波 Low の間で切り換える際に、変調出力信号内の搬送波パルスが短く切り詰められる可能性があります。これを防ぐために、搬送波信号をモジュレータ信号に同期させる事ができます。同期を有効にすると、搬送波信号源切り換えの際に混合する搬送波パルスが Low に遷移してから、DSM は他方の搬送波信号源に切り換わります。

同期は搬送波 High および搬送波 Low の信号源それぞれに対して個別に有効化します。搬送波 High 信号の同期は、MDxCON1 レジスタの CHSYNC ビットをセットすると有効になります。搬送波 Low 信号の同期は、MDxCON1 レジスタの CLSYNC ビットをセットすると有効になります。

図 31-1 から図 31-5 に各種の同期方法を使ったタイミング図を示します。

図 31-2: ON/OFF キーイング (OOK) の同期



例 31-1: 同期なし (MDSHSYNC = 0、MDCLSYNC = 0)

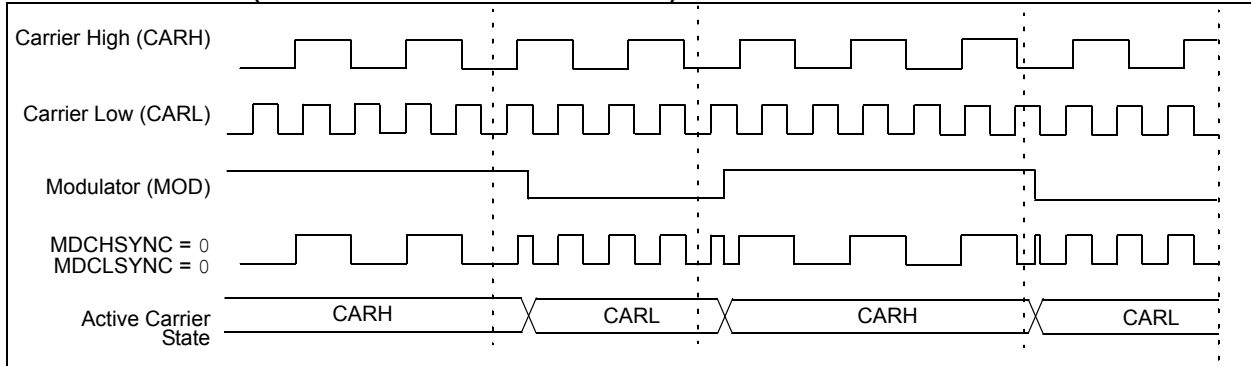
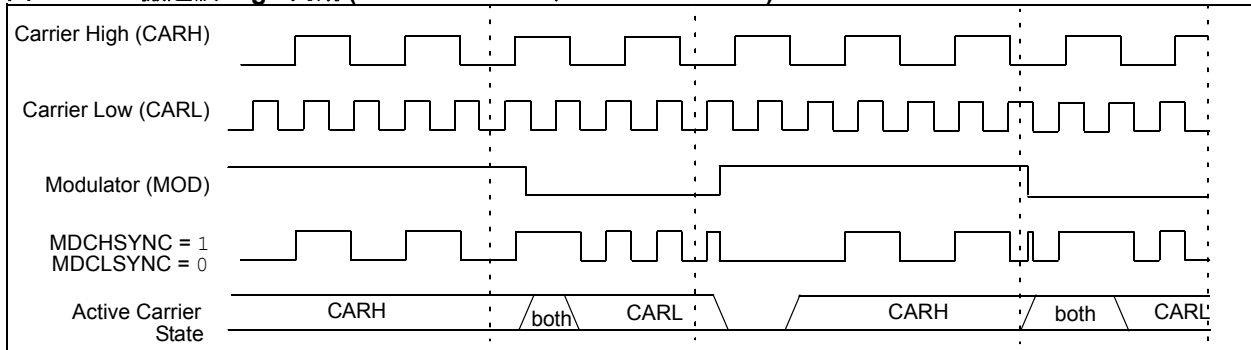


図 31-3: 搬送波 High 同期 (MDSHSYNC = 1、MDCLSYNC = 0)



PIC16(L)F1764/5/8/9

图 31-4: 搬送波 Low 同期 (MDSHSYNC = 0、MDCLSYNC = 1)

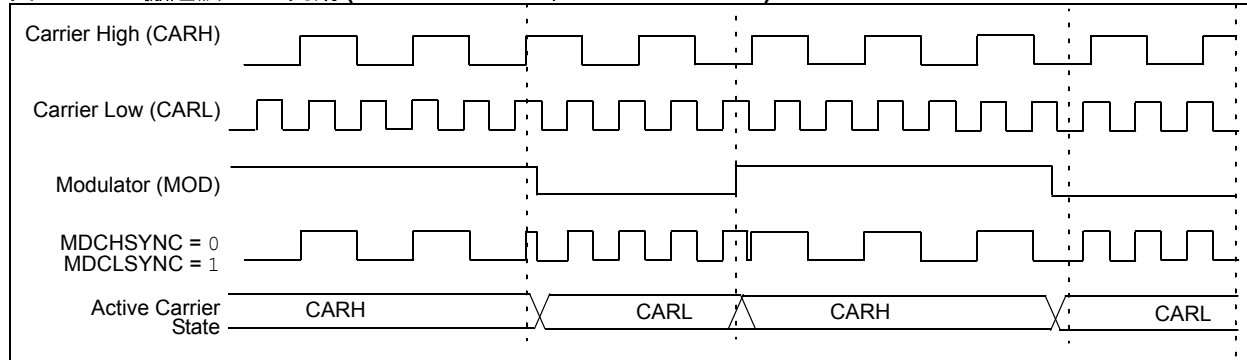
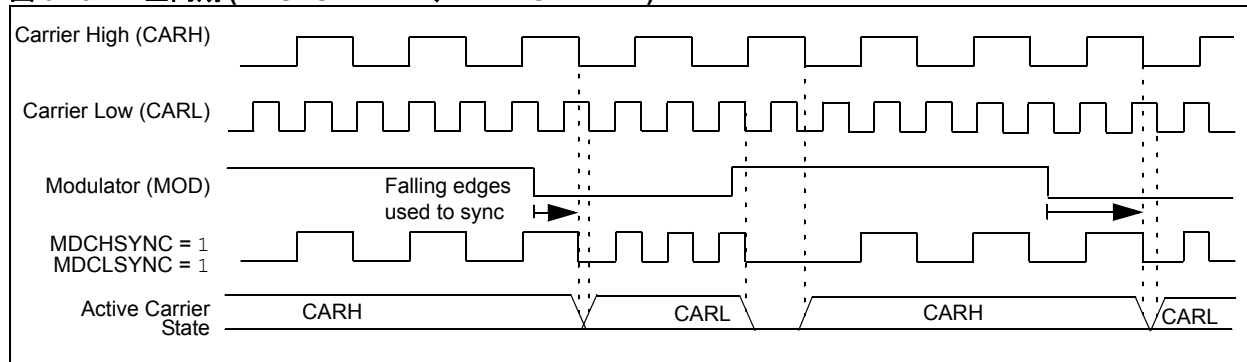


图 31-5: 全同期 (MDSHSYNC = 1、MDCLSYNC = 1)



31.5 ピン間の入力と出力

モジュレータ信号源と搬送波信号源はPPS制御ロジックを使って全てのデバイスピンから選択できます。ピンの選択には以下の2つの設定が必要です。つまり信号源の選択で PPS を使う事を決定し、PPS 制御で目的のピンを選択します。表 31-1 に信号源と PPS レジスタを示します。レジスタ 12-1 とレジスタ 12-2 に PPS レジスタによるピンの選択方法を示します。

表 31-1:

信号源	信号源レジスタ	PPS レジスタ
モジュレータ	MDxSRC	MDxMODPPS
搬送波 High	MDxCARH	MDxCHPPS
搬送波 Low	MDxCARL	MDxCLPPS

個々のピンの PPS 制御を使って、変調出力として全てのデバイスピンを選択できます。ピンにおける出力の選択方法はレジスタ 12-2 を参照してください。

31.6 搬送波信号源の極性選択

搬送波 High および搬送波 Low 信号として選択した入力源からの供給信号は、全て反転できます。MDxCON1 レジスタの CHPOL ビットをセットすると搬送波 High 信号が反転し、CLPOL ビットをセットすると搬送波 Low 信号が反転します。

31.7 プログラマブルなモジュレータ データ

モジュレータ信号源として MDxCON0 レジスタの BIT ビットを選択できます。BIT 信号源を選択すると、ソフトウェアがそれぞれ所定の変調 High および Low 時に BIT ビットをセットおよびクリアする事でモジュレータ信号を生成します。

31.8 変調出力の極性

MDxOUT ピンに供給される変調出力信号も反転できます。変調出力信号の反転は、MDxCON0 レジスタの OPOL ビットをセットすると有効になります。

31.9 スリープ中の動作

DSM モジュールはスリープの影響を受けません。搬送波およびモジュレータ入力源がスリープ中もアクティブであれば、DSM はスリープ中に動作できます。

31.10 リセットの影響

いかなるデバイスリセットもデータ信号モジュレータモジュールを無効にします。リセット後、出力を有効にする前にモジュールを初期化する必要があります。レジスタは既定値にリセットされます。

PIC16(L)F1764/5/8/9

31.11 レジスタ定義：データ信号モジュレータ

表 31-2 に DSM 周辺モジュールの完全ビット名の接頭辞を示します。詳細は [セクション 1.1.2.2 「完全ビット名」](#) を参照してください。

表 31-2:

周辺モジュール	ビット名の接頭辞
DSM1	MD1
DSM2 ⁽¹⁾	MD2

Note 1: PIC16(L)F1768/9 のみです。

レジスタ 31-1: MDxCON0: 変調制御レジスタ 0

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
EN	—	OUT	OPOL	—	—	—	ビット
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **EN:** モジュレータ モジュール イネーブルビット
1 = モジュレータ モジュールを有効にし、入力信号を混合する
0 = モジュレータ モジュールを無効にする (出力なし)
- bit 6 **未実装:** 「0」として読み出し
- bit 5 **OUT:** モジュレータ出力ビット
モジュレータ モジュールの現在の出力値を表示する ⁽¹⁾
- bit 4 **OPOL:** モジュレータ出力極性選択ビット
1 = モジュレータ出力信号を反転する。アイドルで High を出力する
0 = モジュレータ出力信号を反転しない。アイドルで Low を出力する
- bit 3-1 **未実装:** 「0」として読み出し
- bit 0 **BIT:** モジュールへの変調源入力をソフトウェアで直接設定できるようにする ⁽²⁾
1 = モジュレータは搬送波 High 信号源を使う
0 = モジュレータは搬送波 Low 信号源を使う

- Note 1:** 変調された出力の周波数は、このレジスタビットを更新するクロックの周波数よりも大きくかつ非同期である可能性があります。より高速のモジュレータまたは搬送信号では、ビットの値が無効となる場合があります。
- 2:** この動作を実行するには、MDSRC レジスタでモジュレータ信号源として BIT を選択する必要があります。

レジスタ 31-2: MDxCON1: 変調制御レジスタ 1

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-6 **未実装:** 「0」として読み出し

bit 5 **CHPOL:** 変調搬送波 High 極性選択ビット

- 1 = 選択した搬送波 High 信号源を反転する
- 0 = 選択した搬送波 High 信号源を反転しない

bit 4 **CHSYNC:** 変調搬送波 High 同期イネーブルビット

- 1 = モジュレータは搬送波 High 信号の立ち下がリエッジまで待ってから搬送波 Low 信号に切り換える
- 0 = モジュレータ出力を搬送波 High 信号に同期させない⁽¹⁾

bit 3-2 **未実装:** 「0」として読み出し

bit 1 **CLPOL:** 変調搬送波 Low 極性選択ビット

- 1 = 選択した搬送波 Low 信号源を反転する
- 0 = 選択した搬送波 Low 信号源を反転しない

bit 0 **CLSYNC:** 変調搬送波 Low 同期イネーブルビット

- 1 = モジュレータは搬送波 Low 信号の立ち下がリエッジまで待ってから搬送波 High 信号に切り換える
- 0 = モジュレータ出力を搬送波 Low 信号に同期させない⁽¹⁾

Note 1: キャリアに同期していない場合、信号ストリーム内でキャリアパルス幅の欠け、またはグリッチが発生する場合があります。

PIC16(L)F1764/5/8/9

レジスタ 31-3: MDxSRC: モジュレータ信号源制御レジスタ

U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	
—	—	—	MS<4:0>					
bit 7							bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-5 **未実装:** 「0」として読み出し

bit 4-0 **MS<3:0>** モジュレータ信号源選択ビット

表 31-3 を参照してください。

Note 1: キャリアに同期していない場合、信号ストリーム内でキャリアパルス幅の欠け、またはグリッチが発生する場合があります。

表 31-3: モジュレータ信号源

MS<4:0>	モジュレータ信号源 PIC16(L)F1764/5	モジュレータ信号源 PIC16(L)F1768/9
11111-10100	Low 固定	Low 固定
10011	Low 固定	sync_C4OUT
10010	Low 固定	sync_C3OUT
10001	sync_C2OUT	sync_C2OUT
10000	sync_C1OUT	sync_C1OUT
01111	LC3_out	LC3_out
01110	LC2_out	LC2_out
01101	LC1_out	LC1_out
01100	Low 固定	PWM6_out
01011	PWM5_out	PWM5_out
01010	Low 固定	PWM4_out
01001	PWM3_out	PWM3_out
01000	Low 固定	CCP2_out
00111	CCP1_out	CCP1_out
00110	SDO_out	SDO_out
00101	Low 固定	COG2A
00100	DT	DT
00011	TX_out	TX_out
00010	COG1A	COG1A
00001	MDxBIT	MDxBIT
00000	MDxMODPPS ピン選択	MDxMODPPS ピン選択

PIC16(L)F1764/5/8/9

レジスタ 31-4: MDxCARH: 変調搬送波 High 制御レジスタ

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	CH<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **未実装:** 「0」として読み出し

bit 3-0 **CH<3:0>:** モジュレータ データ搬送波 High 選択ビット (1)
表 31-4 を参照してください。

Note 1: キャリアに同期していない場合、信号ストリーム内でキャリアパルス幅の欠け、またはグリッチが発生する場合があります。

表 31-4: 搬送波 High 信号源

CH<3:0>	搬送波 High 信号源 PIC16(L)F1764/5	搬送波 High 信号源 PIC16(L)F1768/9
1111	LC3_out	LC3_out
1110	LC2_out	LC2_out
1101	LC1_out	LC1_out
1100	Low 固定	PWM6_out
1011	PWM5_out	PWM5_out
1010	Low 固定	PWM4_out
1001	PWM3_out	PWM3_out
1000	Low 固定	CCP2_out
0111	CCP1_out	CCP1_out
0110	Low 固定	Low 固定
0101	Low 固定	Low 固定
0100	Low 固定	Low 固定
0011	Low 固定	Low 固定
0010	HFINTOSC	HFINTOSC
0001	Fosc	Fosc
0000	MDxCHPPS ピン選択	MDxCHPPS ピン選択

PIC16(L)F1764/5/8/9

レジスタ 31-5: MDxCARL: 変調搬送波 Low 制御レジスタ

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	CL<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

bit 7-4 **未実装:** 「0」として読み出し
bit 3-0 **CL<3:0>** モジュレータ データ搬送波 Low 選択ビット (1)
表 31-5 を参照してください。

Note 1: キャリアに同期していない場合、信号ストリーム内でキャリアパルス幅の欠け、またはグリッチが発生する場合があります。

表 31-5: 搬送波 Low 信号源

CL<3:0>	搬送波 Low 信号源 PIC16(L)F1764/5	搬送波 Low 信号源 PIC16(L)F1768/9
1111	LC3_out	LC3_out
1110	LC2_out	LC2_out
1101	LC1_out	LC1_out
1100	Low 固定	PWM6_out
1011	PWM5_out	PWM5_out
1010	Low 固定	PWM4_out
1001	PWM3_out	PWM3_out
1000	Low 固定	CCP2_out
0111	CCP1_out	CCP1_out
0110	Low 固定	Low 固定
0101	Low 固定	Low 固定
0100	Low 固定	Low 固定
0011	Low 固定	Low 固定
0010	HFINTOSC	HFINTOSC
0001	Fosc	Fosc
0000	MDxCLPPS ピン選択	MDxCLPPS ピン選択

表 31-6: データ信号モジュレータ モード関連のレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
MDxCARH	—	—	—	—	CH<3:0>				373
MDxCARL	—	—	—	—	CL<3:0>				374
MDxSRC	—	—	—	MS<4:0>					371
MDxCON0	EN	—	OUT	OPOL	—	—	—	ビット	370
MDxCON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	370

凡例: — = 未実装、「0」として読み出し。網掛けの部分はデータ信号モジュレータでは使いません。

32.0 マスタ同期シリアルポート (MSSP) モジュール

32.1 MSSP モジュールの概要

マスタ同期シリアルポート (MSSP) モジュールは、他の周辺デバイスまたはマイクロコントローラとの通信に便利なシリアルインターフェイスです。周辺デバイスとはシリアル EEPROM、シフトレジスタ、ディスプレイドライバ、A/D コンバータ (ADC) 等です。MSSP モジュールは、以下の2つのモードのいずれかで動作します。

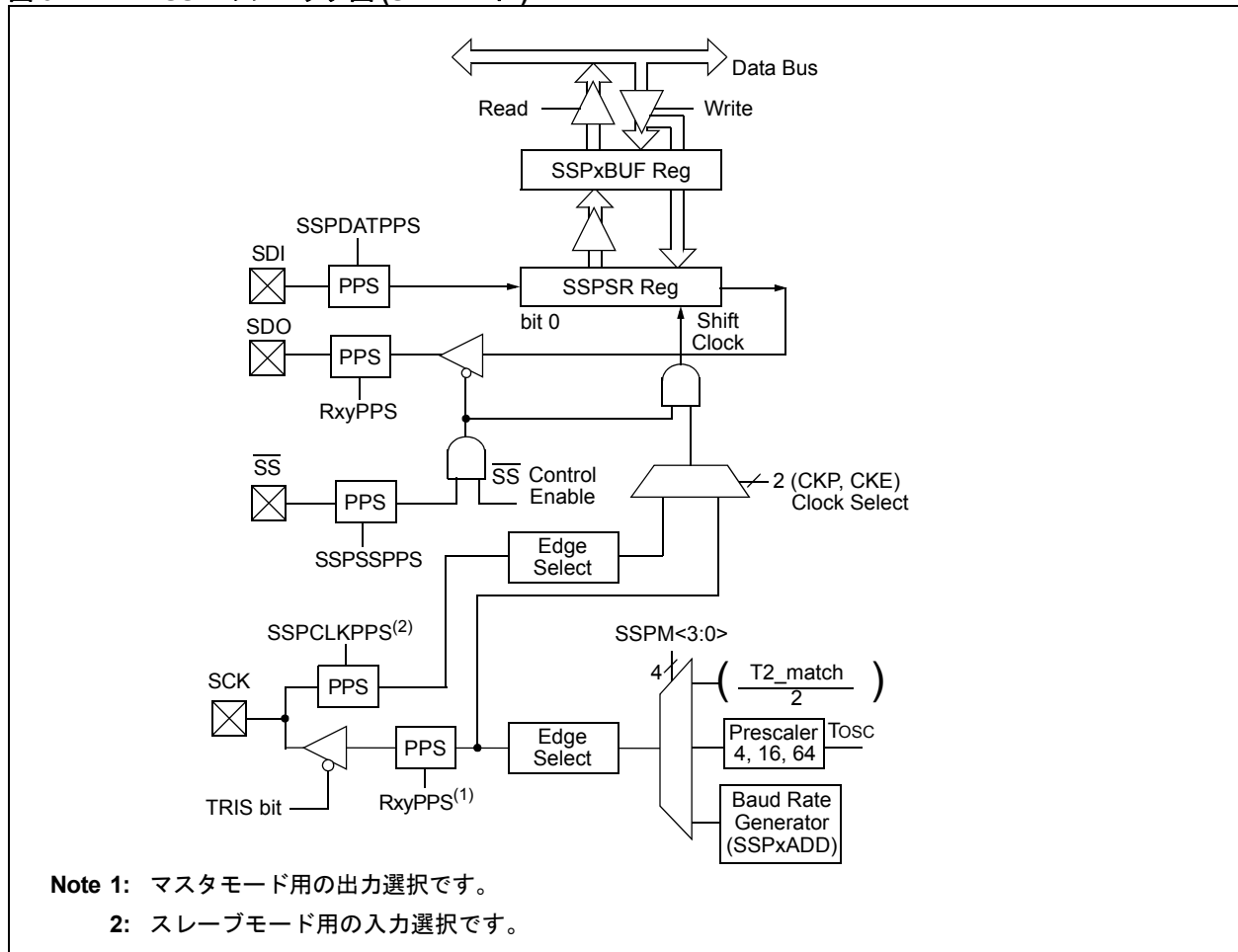
- シリアル ペリフェラル インターフェイス (SPI)
- I²C™ (Inter-Integrated Circuit)

SPI インターフェイスは、以下のモードと機能をサポートします。

- マスタモード
- スレーブモード
- クロックパリティ
- スレーブセレクト同期 (スレーブモードのみ)
- スレーブデバイスのデジチェーン接続

図 32-1 に、SPI インターフェイス モジュールのブロック図を示します。

図 32-1: MSSP のブロック図 (SPI モード)



PIC16(L)F1764/5/8/9

I²Cインターフェイスは、以下のモードと機能をサポートします。

- マスタモード
- スレーブモード
- バイト単位の NACK(スレーブモード)
- マルチマスタ(制約あり)
- 7/10ビットアドレス指定
- スタート/ストップ割り込み
- 割り込みマスク
- クロックストレッチ
- バスコリジョンの検出
- ジェネラルコールアドレスマッチング
- アドレスマスク
- アドレス/データホールドモード
- SDAホールド時間の選択

図 32-2 にマスタモード I²C インターフェイスのブロック図を示します。図 32-3 にスレーブモード I²C インターフェイスのブロック図を示します。

図 32-2: MSSP のブロック図 (I²C™ マスタモード)

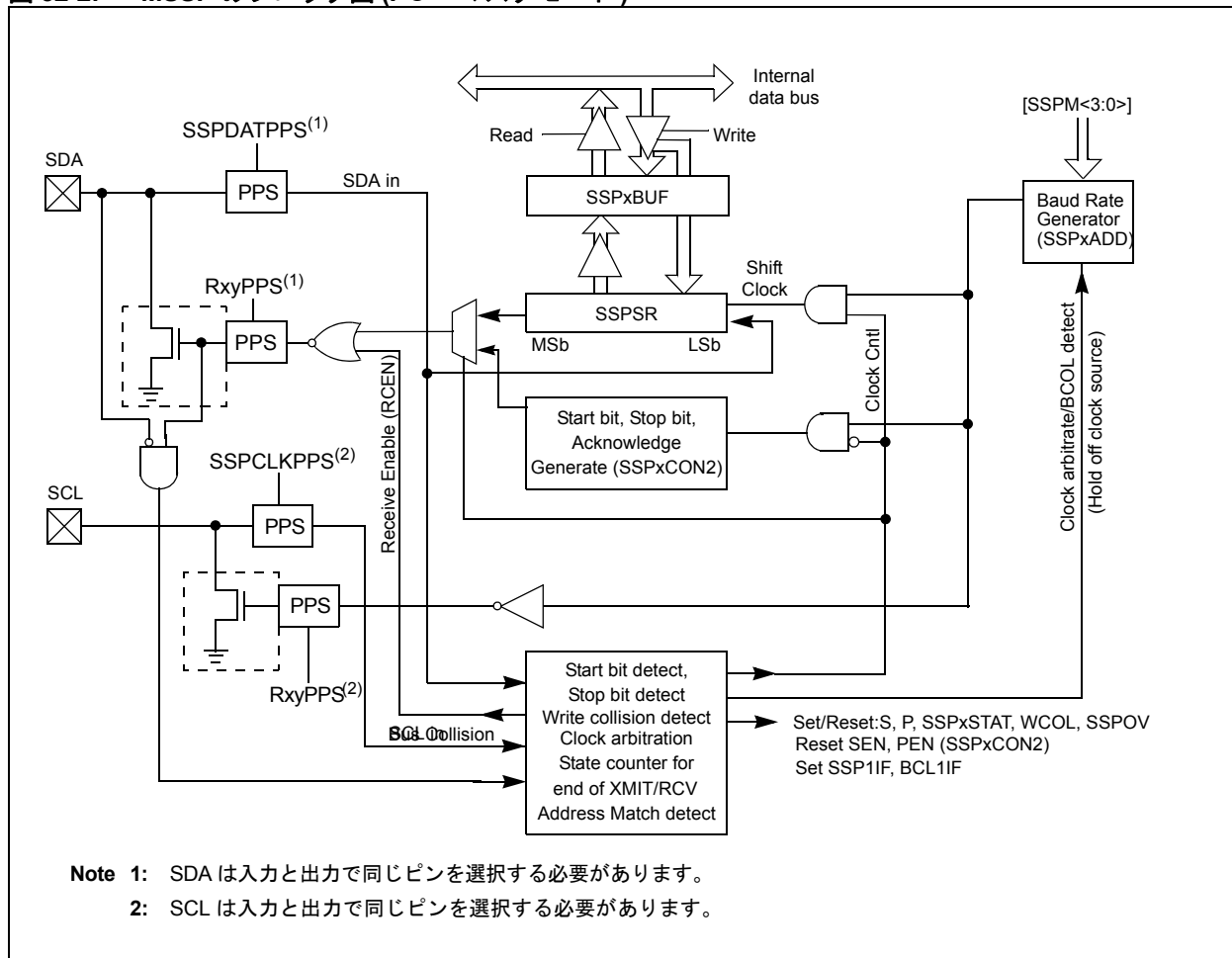
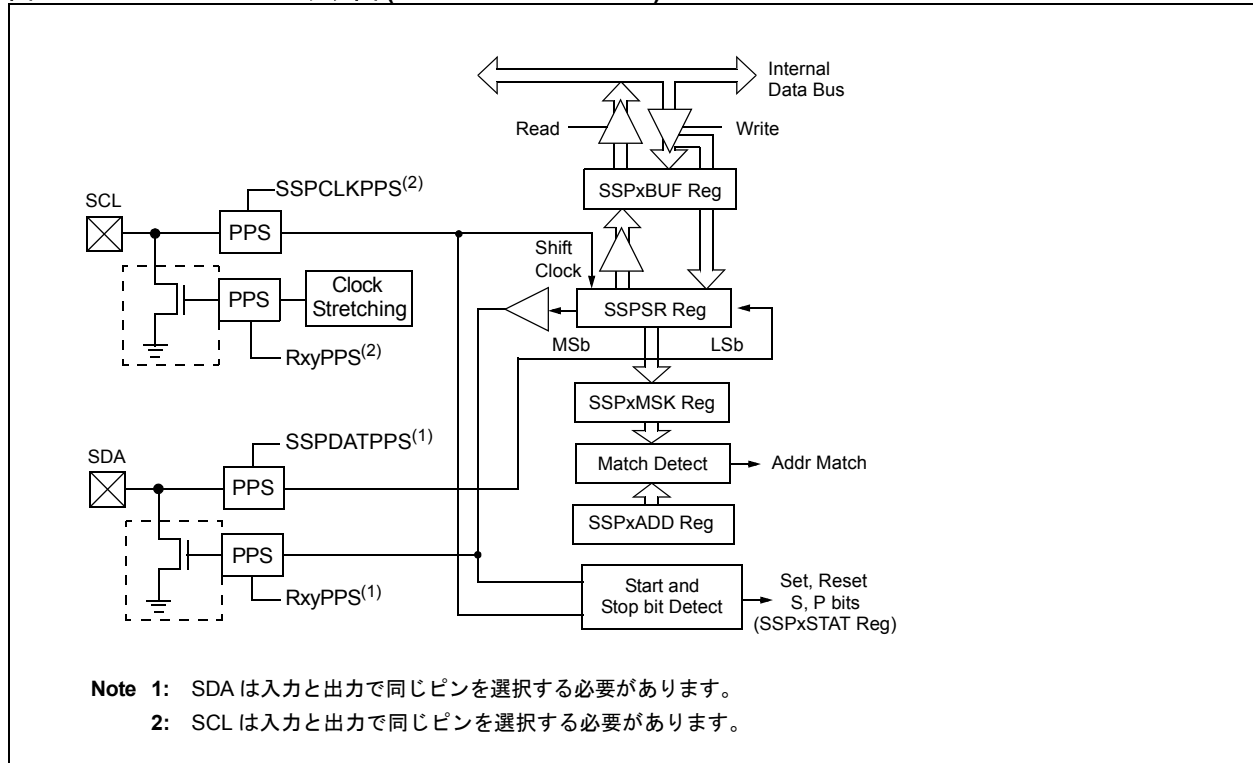


図 32-3: MSSSP のブロック図 (I²C™ スレーブモード)



32.2 SPI モードの概要

SPI(シリアル ペリフェラル インターフェイス) バスは、全二重モードで動作する同期シリアルデータ通信バスです。デバイスはマスタ / スレーブ環境で通信し、マスタデバイスから通信を開始します。スレーブデバイスは、スレーブセレクトと呼ばれるチップセレクト信号で制御されます。

SPI バスでは、以下の 4 つの信号接続を使います。

- シリアルクロック (SCK)
- シリアルデータ出力 (SDO)
- シリアルデータ入力 (SDI)
- スレーブセレクト (SS)

図 32-1 に、SPI モードで動作する MSSP モジュールのブロック図を示します。

SPI バスは、1 つのマスタデバイスと 1 つまたは複数のスレーブデバイス間で動作します。複数のスレーブデバイスを使う場合、マスタデバイスから各スレーブデバイスにそれぞれ独立したスレーブセレクト接続が必要です。

図 32-4 にマスタデバイスと複数のスレーブデバイス間の代表的な接続を示します。

マスタは一度に 1 つだけスレーブを選択します。ほとんどのスレーブデバイスの出力は 3 ステートであり、非選択時はその出力信号がバスから切り離されているように見えます。

転送には 8 ビットのサイズのシフトレジスタを、マスタとスレーブで 1 つずつ使います。マスタまたはスレーブデバイスのどちらでも、データは常に一度に 1 ビットずつ、最上位ビット (MSb) から順にシフト出力されます。同時に、新しい最下位ビット (LSb) が同じレジスタにシフト入力されます。

図 32-5 に、マスタおよびスレーブデバイスとして設定された 2 つのプロセッサ間の代表的な接続を示します。

データは、プログラムされたクロックエッジで両方のシフトレジスタからシフト出力され、逆のクロックエッジでラッチされます。

マスタデバイスは SDO 出力ピンから情報を送信します。このピンはスレーブの SDI 入力ピンへと接続され、スレーブはこのピンから情報を受信します。スレーブデバイスは SDO 出力ピンから情報を送信します。このピンはマスタの SDI 入力ピンへと接続され、マスタはこのピンから情報を受信します。

通信開始時、まずマスタデバイスがクロック信号を送信します。マスタとスレーブの両デバイスは、同じクロック極性に設定する必要があります。

マスタデバイスはシフトレジスタから MSb を送信する事で転送を開始します。スレーブデバイスは、同じラインからこのビットを読み出し、自身のシフトレジスタの LSb 位置に保存します。

SPI の各クロックサイクルでは、全二重でデータが転送されます。つまり、マスタデバイスのシフトレジスタから MSb が SDO ピンを介して送信されてスレーブデバイスのシフトレジスタの LSb 位置に読み込まれ、スレーブデバイスのシフトレジスタからも MSb が SDO ピンを介して送信されてマスタデバイスのシフトレジスタの LSb 位置に読み込まれるという事を意味します。

8 ビット分のデータがシフト出力された時点で、マスタとスレーブのレジスタ値交換が終了します。

さらに交換すべきデータが存在する場合、シフトレジスタに新しいデータが書き込まれ、上記のプロセスが繰り返されます。

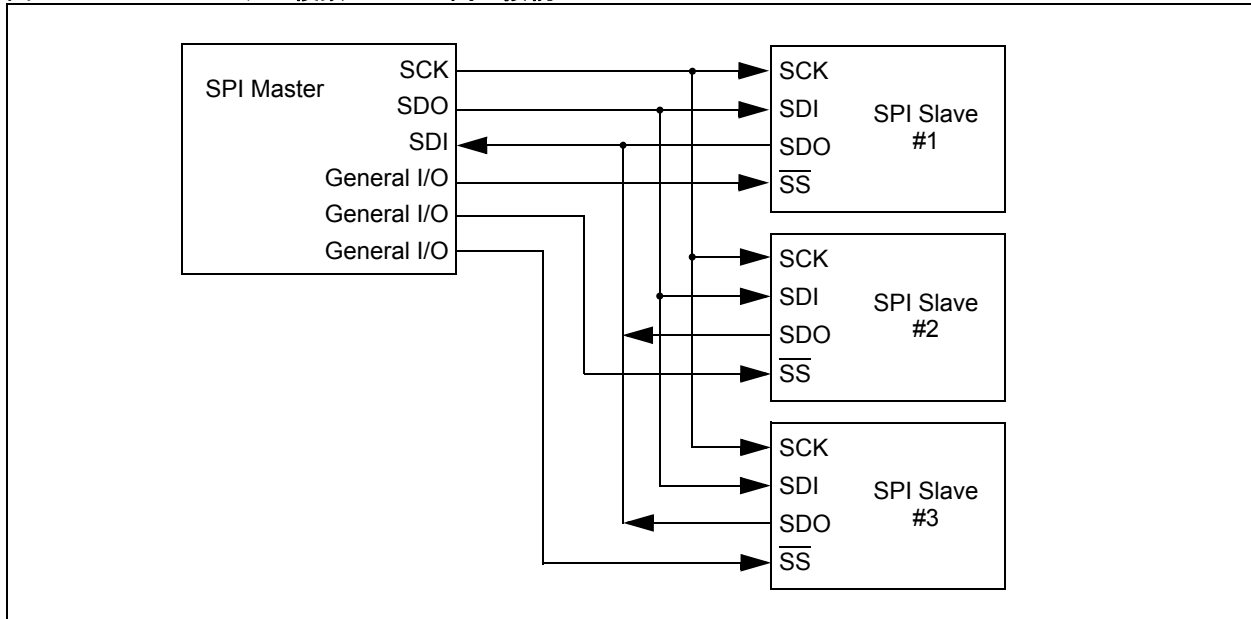
そのデータが意味を持つか、またはダミーデータであるかは、アプリケーション ソフトウェアで決まります。データ転送における 3 つのシナリオは以下の通りです。

- マスタが意味のあるデータを送信し、スレーブがダミーデータを送信する。
- マスタが意味のあるデータを送信し、スレーブも意味のあるデータを送信する。
- マスタがダミーデータを送信し、スレーブが意味のあるデータを送信する。

送信に使うクロックサイクル数には、特に制限はありません。送信すべきデータがなくなると、マスタがクロック信号の送信を停止し、スレーブの選択を解除します。

バスに接続されていてもスレーブセレクトラインで選択されていないスレーブデバイスは、クロックと転送信号を無視し、データ送信もできません。

図 32-4: SPI マスタと複数スレーブ間の接続



32.2.1 SPI モードレジスタ

SPIモード動作では以下の5つのレジスタを使います。これらのレジスタには、以下のようなものがあります。

- MSSP STATUS レジスタ (SSPxSTAT)
- MSSP 制御レジスタ 1 (SSPxCON1)
- MSSP 制御レジスタ 3 (SSPxCON3)
- MSSP データバッファ レジスタ (SSPxBUF)
- MSSP アドレスレジスタ (SSPxADD)
- MSSPシフトレジスタ (SSPSR) (直接アクセス不可)

SSPxCON1 と SSPxSTAT は、SPI モード動作時の制御レジスタと STATUS レジスタです。SSPxCON1 レジスタは、読み書き可能です。SSPxSTAT の下位 6 ビットは読み出し専用です。SSPxSTAT の上位 2 ビットは読み書き可能です。

SPI マスタモードの場合、SSPxADD に baud レートジェネレータで使われる値を書き込みます。baud レートジェネレータの詳細は、[セクション 32.7「baud レートジェネレータ」](#)を参照してください。

SSPSR は、データのシフト入出力に使うシフトレジスタです。SSPxBUF を使うと、SSPSR レジスタに間接的にアクセスできます。SSPxBUF は、データバイトを読み書きするバッファレジスタです。

受信動作では、SSPSR と SSPxBUF でバッファ付きのレシーバを構成します。SSPSR が 1 バイト分のデータ全体を受信すると、そのバイトは SSPxBUF に転送され、SSPxIF 割り込みがセットされます。

転送時、SSPxBUF はバッファリングされません。SSPxBUF に書き込むと、SSPxBUF と SSPSR の両方に書き込まれます。

32.2.2 SPI モードの動作

SPI の初期化には、複数の設定が必要です。対応する制御ビット (SSPxCON<15:0> と SSPxSTAT<7:6>) に適切な値を書き込む事で、それらのオプションを設定します。これらの制御ビットで以下の項目を設定できます。

- マスタモード (SCK はクロック出力)
- スレーブモード (SCK はクロック入力)
- クロック極性 (SCK のアイドル状態)
- データ入力サンプリング位相 (データ出力期間の間または最後)
- クロックエッジ (SCK の立ち上がり/立ち下がりエッジでデータを出力)
- クロックレート (マスタモードのみ)
- スレーブセレクトモード (スレーブモードのみ)

シリアルポートを有効にするには、SSP イネーブルビット (SSPxCON1 レジスタの SSPEN) をセットする必要があります。SPI モードをリセットまたは再設定する場合、SSPEN ビットをクリアし、SSPxCONx レジスタを再度初期化した後で SSPEN ビットをセットします。これによって、SDI、SDO、SCK、 \overline{SS} ピンがシリアルポートピンとして設定されます。シリアルポートとして動作するピンの中には、データ方向ビット (TRIS レジスタ) を以下の通りプログラムする必要があります。

- SDI: 対応する TRIS ビットをセットする
- SDO: 対応する TRIS ビットをクリアする
- SCK (マスタモード): 対応する TRIS ビットをクリアする
- SCK (スレーブモード): 対応する TRIS ビットをセットする
- \overline{SS} : 対応する TRIS ビットをセットする

PIC16(L)F1764/5/8/9

シリアルポート機能の中に不要なものがあれば、対応するデータ方向レジスタ (TRIS) を逆の値にプログラムする事で無効にできます。

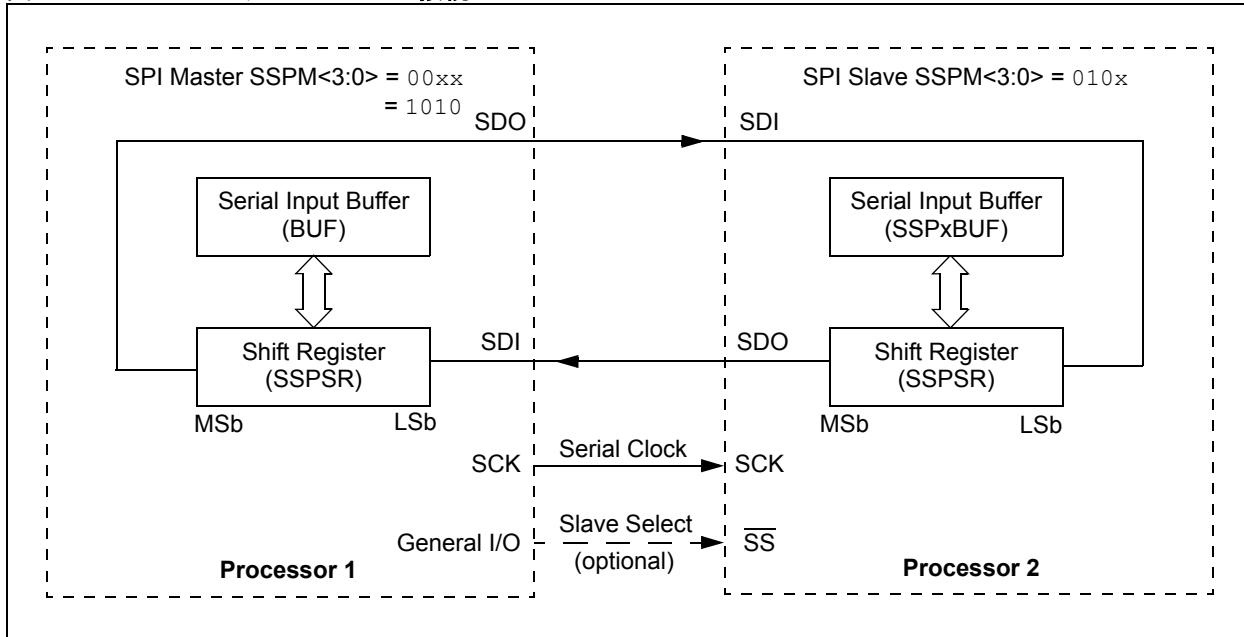
MSSP は送信 / 受信シフトレジスタ (SSPSR) とバッファレジスタ (SSPxBUF) で構成されます。SSPSR は MSb から順にデータをシフト入力またはシフト出力します。SSPxBUF は受信データが揃うまで、SSPSR に書き込まれたデータを保持します。8 ビットのデータの受信が完了すると、受信したバイトは SSPxBUF レジスタに転送されます。そして SSPxSTAT レジスタのバッファフル検出ビット BF と、割り込みフラグビット SSPxIF がセットされます。このようなダブルバッファ方式 (SSPxBUF) でデータを受信する事で、受信済みデータを読み出す前に次のバイトの受信を開始できます。データ送受信中の SSPxBUF レジスタへの書き込みは全て無視され、SSPxCON1 レジスタの書き込みコリジョン検出ビット WCOL がセットさ

れます。コリジョン発生後に SSPxBUF レジスタへの書き込みが適切に完了するように、ユーザソフトウェアで WCOL ビットをクリアする必要があります。

アプリケーションソフトウェアが有効な受信データを必要としている場合、次の転送バイトデータを SSPxBUF へ書き込む前に SSPxBUF のデータを読み出す必要があります。SSPxBUF に受信データが書き込まれると (転送完了)、SSPxSTAT レジスタのバッファフルビット (BF) がセットされます。BF ビットは SSPxBUF を読み出すとクリアされます。SPI が送信のみの場合、このデータは無意味である場合があります。通常、送受信の完了を判断するには、MSSP 割り込みを使います。割り込みによる方法を使わずに書き込みコリジョンを避けるには、ソフトウェアポーリングを実行します。

SSPSR は直接読み書きできません。アクセスするには SSPxBUF レジスタのアドレスを指定します。また、各種のステータス条件は SSPxSTAT レジスタによって示されます。

図 32-5: SPI マスタ / スレーブの接続



32.2.3 SPI マスタモード

マスタは SCK ラインを制御するため、いつでもデータ転送を開始できます。また、マスタはソフトウェアプロトコルでスレーブ (図 32-5 の Processor 2) がデータをブロードキャストするタイミングを決定します。

マスタモードでは、SSPxBUF レジスタに書き込むと、データはただちに送信 / 受信されます。SPI を受信にのみ使う場合、SDO 出力を無効に (入力としてプログラム) してかまいません。SSPSR レジスタは、プログラムされたクロック速度で SDI ピンの信号のシフト入力を継続します。1 バイト受信するごとに、通常の受信バイトと同じように SSPxBUF レジスタに読み込みます (割り込みとステータスビットも必要に応じてセットされます)。

クロック極性は SSPxCON1 レジスタの CKP ビットと SSPxSTAT レジスタの CKE ビットで設定します。これによって 図 32-6、図 32-8、図 32-9、図 32-10 に示す SPI 通信の波形が得られます。この通信では MSb が最初に送信されています。マスタモードでは、SPI のクロック速度 (ビットレート) を以下のいずれかに設定できます。

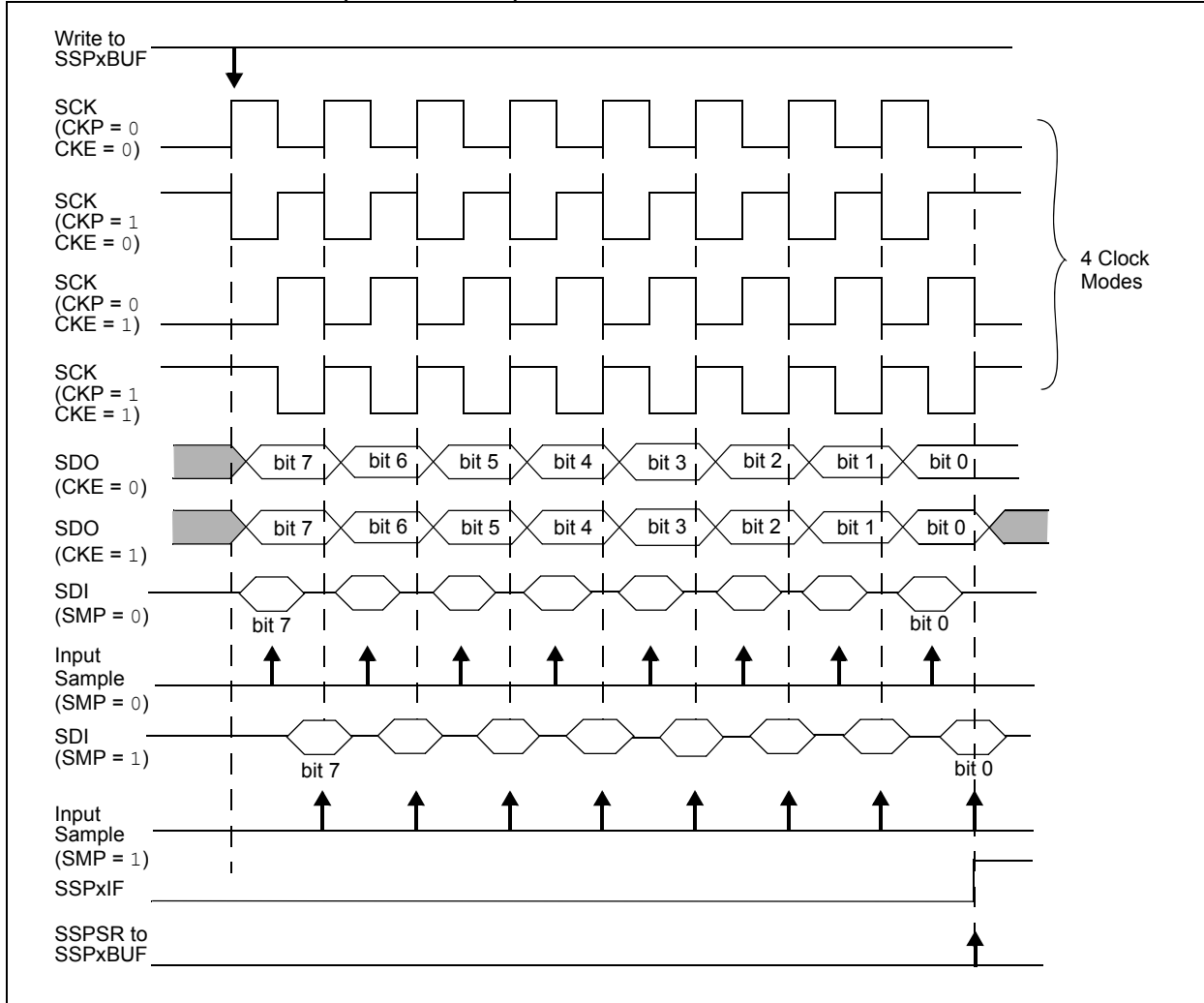
- $F_{osc}/4$ (または T_{cy})
- $F_{osc}/16$ (または $4 * T_{cy}$)
- $F_{osc}/64$ (または $16 * T_{cy}$)
- Timer2 出力 /2
- $F_{osc}/(4 * (SSPxADD + 1))$

図 32-6 に、マスタモードの波形を示します。

CKE ビットをセットした場合、SDO のデータは SCK にクロックエッジが表れる前に有効になります。この図には、SMP ビットの状態に応じた入力サンプリング位置の変化が示してあります。また、受信データが SSPxBUF に書き込まれるタイミングも示しています。

Note: マスタモードでは、SCK ピンに出力されるクロック信号をこのモジュールの入力クロック信号としても使います。RxyPPS レジスタで出力用に選択されたピンも SSPCLKPPS レジスタで周辺モジュールの入力として選択する必要があります。

図 32-6: SPI モードの波形 (マスタモード)



32.2.4 SPI スレーブモード

スレーブモードでは、SCK に外部クロックパルスが印加されるとデータが送受信されます。最後のビットがラッチされると、SSPxIF 割り込みフラグビットがセットされます。

モジュールを SPI スレーブモードで動作させる前に、クロックラインのアイドル状態を一致させておく必要があります。クロックラインの状態は SCK ピンを読み出す事で調べられます。アイドル状態は、SSPxCON1 レジスタの CKP ビットで決まります。

スレーブモードでは、SCK ピンに外部クロック源からのクロックが供給されます。この外部クロックは、電気的仕様で指定されている最小 High/Low 期間を満たす必要があります。

スリープ中でもスレーブはデータを送受信できます。シフトレジスタには SCK ピン入力からクロックが供給され、1 バイトを受信するとデバイスが割り込みを生成します。割り込み機能が有効の場合、デバイスはスリープから復帰します。

32.2.4.1 デイジーチェーン構成

SPI バスは、デイジーチェーン構成が可能です。最初のスレーブの出力を 2 番目のスレーブの入力に、2 番目のスレーブの出力を 3 番目のスレーブの入力に、という方式で接続します。最後のスレーブの出力をマスタの入力に接続します。各クロックパルス群で、各スレーブは直前のクロックパルス群で受信したデータの正確なコピーを送信します。チェーン全体が 1 つの大きな通信シフトレジスタであるかのように動作します。デイジーチェーン機能は、マスタデバイスからのスレーブセレクトラインを 1 本しか必要としません。

図 32-7 に、SPI モードで動作する代表的なデイジーチェーン接続のブロック図を示します。

デイジーチェーン構成の場合、スレーブで必要なのはバス上にある最新のバイトのみです。SSPxCON3 レジスタの BOEN ビットをセットすると、前のバイトが SSPxBUF レジスタから読み出されていない場合でも SSPxBUF への書き込みが可能です。これによって、ソフトウェアは適用されないデータを無視できます。

32.2.5 スレーブセレクト同期

スレーブセレクトは通信の同期にも使えます。マスタデバイスの通信準備が整うまでスレーブセレクトラインは High に保持されます。スレーブセレクトラインが Low に駆動される事で、スレーブは転送開始を認識します。

スレーブは、通信を適切に受信できなかった場合、転送の最後にスレーブセレクトラインが High に戻る時点でリセットされます。スレーブはその後、次にスレーブセレクトラインが Low に駆動された時に新しい転送を受信できるように待機します。スレーブセレクトラインを使わないと、スレーブがマスタとの同期を失う恐れがあります。この場合、スレーブが1ビットを受信し損なうと、その後の転送でも常に1ビットのずれが生じます。スレーブセレクトラインを使うと、スレーブとマスタは各転送の開始時に毎回同期を取ることができます。

\overline{SS} ピンは同期スレーブモードを実現します。SPI は、 \overline{SS} ピン制御が有効 ($SSPxCON1<3:0> = 0100$) に設定されたスレーブモードである必要があります。

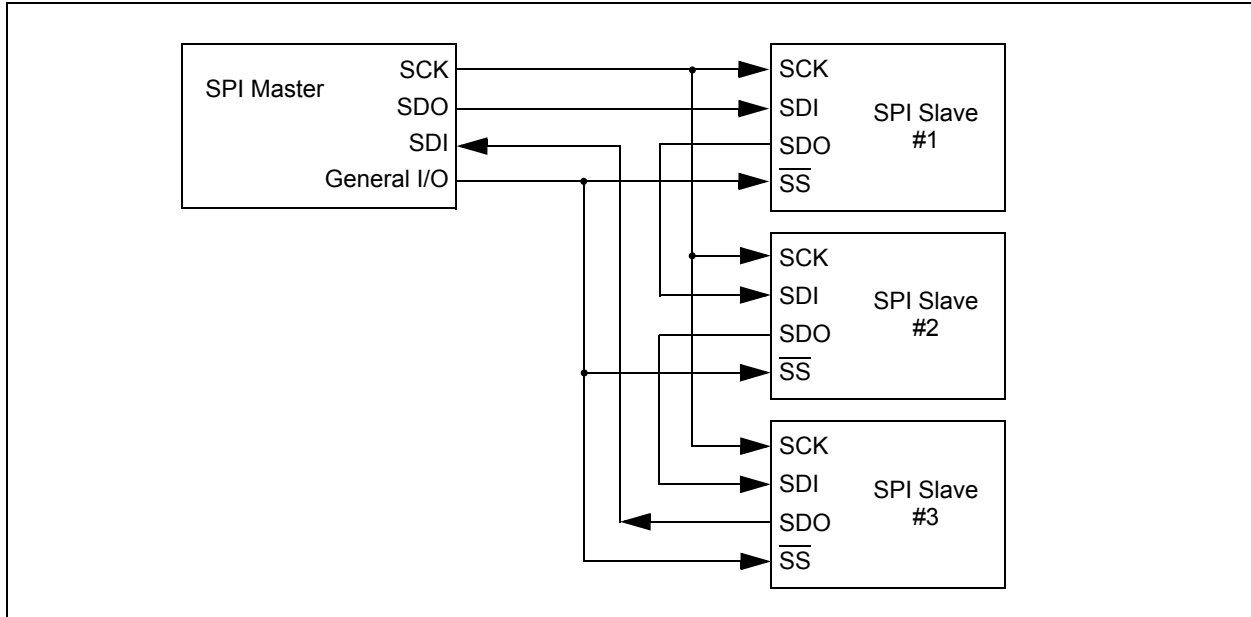
\overline{SS} ピンを Low にすると、送受信が有効になり、SDO ピンが駆動されます。

\overline{SS} ピンを High にすると、たとえバイト送信の途中であっても SDO ピンの駆動は中断され、出力はフローティング状態になります。従って、アプリケーションによっては外付けのプルアップ / プルダウン抵抗の接続が望ましい場合があります。

- Note 1:** SPI が \overline{SS} ピン制御有効 ($SSPxCON1<3:0> = 0100$) のスレーブモードである場合、 \overline{SS} ピンが VDD に設定されると SPI モジュールはリセットされます。
- 2:** SPI が CKE をセットしたスレーブモードの場合、ユーザは \overline{SS} ピン制御を有効にする必要があります。
- 3:** SPI スレーブモードで動作している間、 $SSPxSTAT$ レジスタの SMP ビットはクリア状態を保持する必要があります。

SPI モジュールがリセットされると、ビットカウンタは「0」に設定されます。これは \overline{SS} ピンを強制的に High レベルにするか、 $SSPEN$ ビットをクリアする事で可能です。

図 32-7: SPI デイジーチェーン接続



PIC16(L)F1764/5/8/9

図 32-8: スレーブセレクト同期の波形

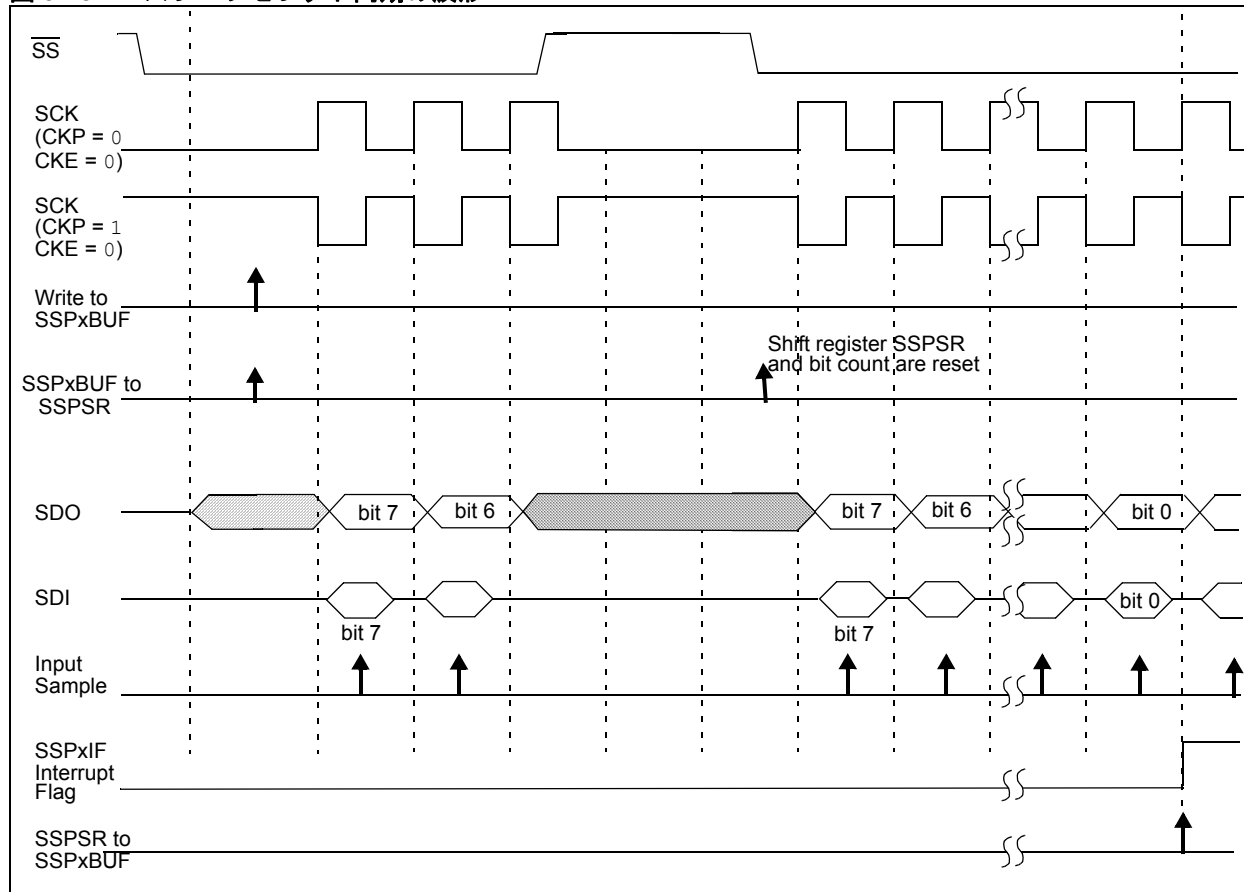


図 32-9: SPI モードの波形 (CKE = 0 のスレーブモード)

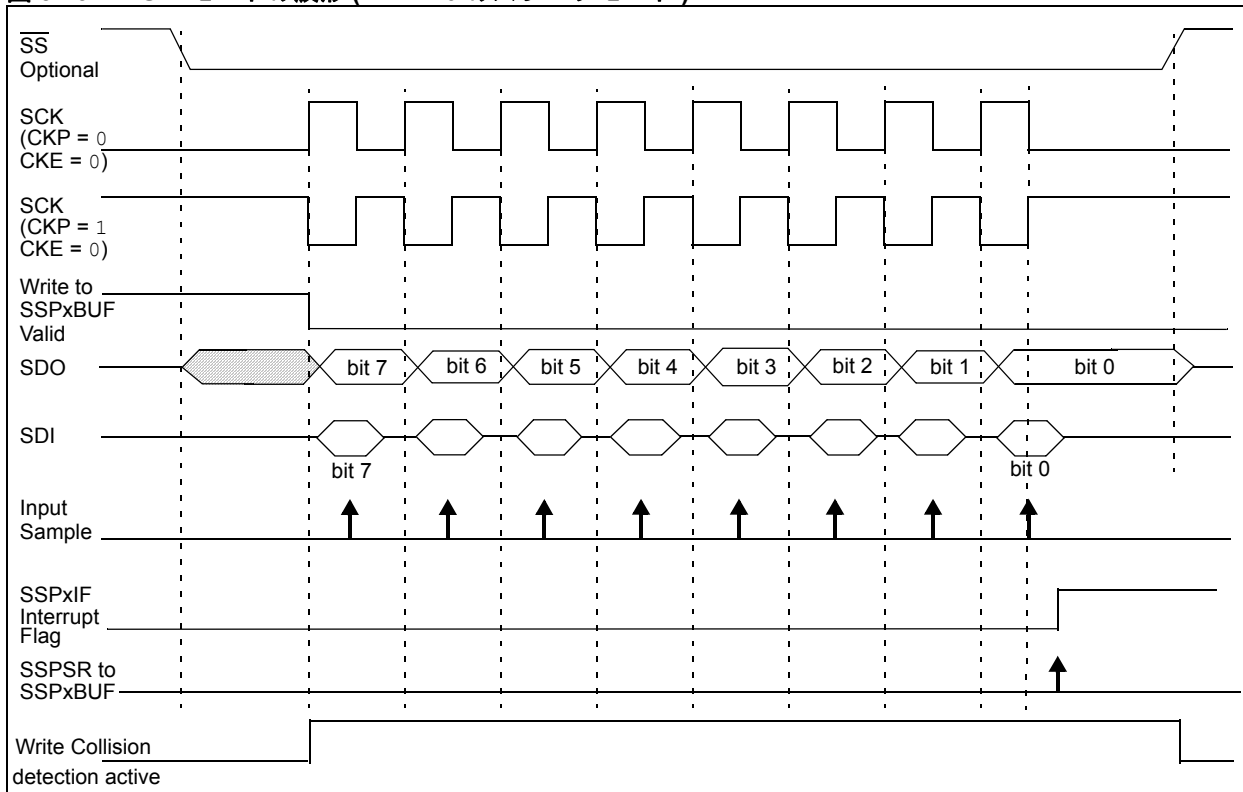
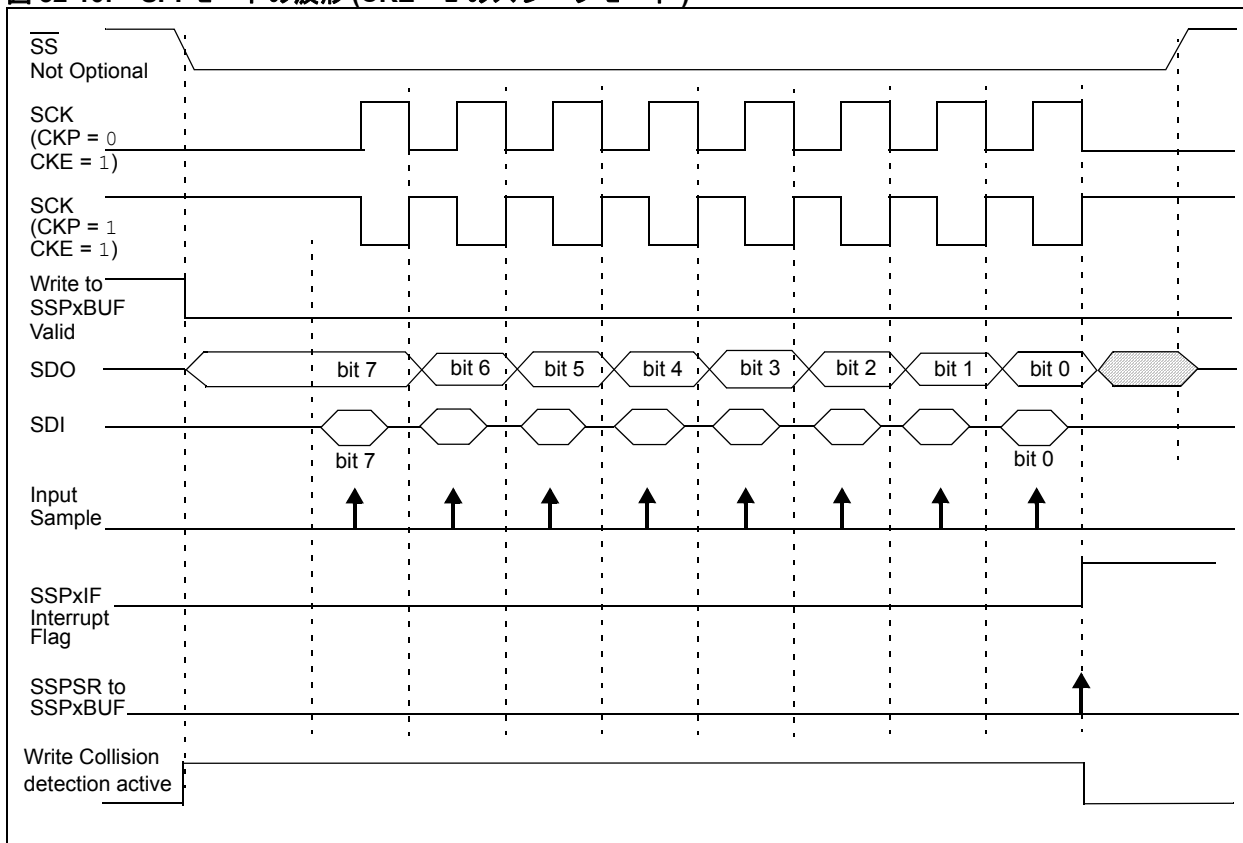


図 32-10: SPI モードの波形 (CKE = 1 のスレーブモード)



PIC16(L)F1764/5/8/9

32.2.6 スリープ中の SPI 動作

SPI マスタモードではモジュール クロックがフルパワーモードとは異なる速度で動作する場合があります。スリープ中は全てのクロックが停止します。

MSSPクロックがシステムクロックよりも大幅に高速の場合、特に注意する必要があります。

スレーブモードでMSSP割り込みを有効にしている場合、マスタがデータ送信を完了すると、MSSP 割り込みによってコントローラがスリープから復帰します。

スリープを維持する場合、MSSP 割り込みを無効にします。

SPI マスタモードでスリープに移行すると、全てのモジュール クロックが停止し、デバイスが復帰するまで送受信はそのままの状態を保持します。デバイスが通常動作に戻ると、モジュールはデータの送受信を再開します。

SPI スレーブモードの場合、SPI 送受信シフトレジスタはデバイスと非同期で動作します。このため、デバイスがスリープに移行しても SPI 送受信シフトレジスタへのデータのシフト入力を継続できません。8 ビットを全て受信すると、MSSP 割り込みフラグビットがセットされ、割り込み機能が有効の場合はデバイスが復帰します。

表 32-1: SPI 動作関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RxyPPS	—	—	—	RxyPPS<4:0>					158
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					158、160
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					158、160
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					158、160
SSP1BUF	同期シリアルポート受信 / 送信バッファレジスタ								379*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				424
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	423
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	423
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例： — = 未実装、「0」として読み出し。網掛けの部分は SPI モードの MSSP では使用しません。

* 本文中でレジスタを説明しているページです。

Note 1: 未実装、「1」として読み出します。

2: PIC16(L)F1768/9 のみです。

32.3 I²C モードの概要

I²C (Inter-Integrated Circuit)バスはマルチマスタ シリアルデータ通信バスです。デバイスはマスタ / スレーブ環境で通信し、マスタデバイス側から通信を開始します。スレーブデバイスはアドレス指定で制御されます。

I²Cバスでは、以下の2つの信号接続を使います。

- シリアルクロック (SCL)
- シリアルデータ (SDA)

図 32-11 に、I²C モードで動作する MSSP モジュールのブロック図を示します。

SCL 接続と SDA 接続はどちらも双方向のオープンドレインラインであり、それぞれに電源電圧へのプルアップ抵抗が必要です。ラインをグラウンドに接続するのは論理 0 で、ラインをフローティングにするのは論理 1 と見なされます。

図 32-11 に、マスタおよびスレーブデバイスとして設定された 2 つのプロセッサ間の代表的な接続を示します。

I²Cバスには、1つまたは複数のマスタデバイスと、1つまたは複数のスレーブデバイスを接続できます。

各デバイスは以下の4つのモードのいずれかで動作します。

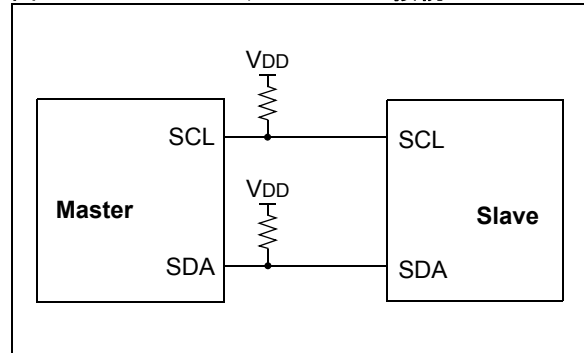
- マスタ送信モード
(マスタがスレーブにデータを送信)
- マスタ受信モード
(マスタがスレーブからデータを受信)
- スレーブ送信モード
(スレーブがマスタにデータを送信)
- スレーブ受信モード
(スレーブがマスタからデータを受信)

通信開始時、マスタデバイスはマスタ送信モードです。マスタデバイスはスタートビットを送り、続いて通信相手のスレーブのアドレスバイトを送ります。次に、1ビットの読み書き (R/W) ビットを送ります。このビットは、マスタがスレーブデバイスにデータを送信するのか、スレーブデバイスからデータを受信するのかを決定します。

マスタが指定したスレーブがバス上に存在する場合、そのスレーブは ACK ビットを返します。その後マスタは送信または受信モードの動作を継続し、スレーブもそれに合わせて受信または送信モードの動作を継続します。

スタートビットは、SCL ラインを High に保ったまま、SDA ラインを High から Low に遷移させる事で示されます。アドレスバイトとデータバイトは、最上位ビット (MSb) から順に送信されます。マスタがスレーブからデータを読み出す場合は読み書きビットとして論理 1 を、スレーブにデータを書き込む場合は論理 0 を送信します。

図 32-11: I²C マスタ / スレーブ接続



ACK ビットはアクティブ Low 信号であり、スレーブデバイスは送信データの受信が完了して後続データの受信準備ができたなら SDA ラインを Low に保持してトランスミッタに伝えます。

データビットの遷移は、必ず SCL ラインが Low に保持されている間に行われます。SCL ラインが High に保持されている間の遷移は、スタートビットとストップビットを表すために使います。

マスタがスレーブに書き込む場合、マスタがデータを 1 バイト送信するとスレーブが ACK ビットで応答するという動作を繰り返します。この場合、マスタデバイスはマスタ送信モードで、スレーブデバイスはスレーブ受信モードで動作します。

マスタがスレーブから読み出す場合、マスタはスレーブからデータを 1 バイト受信すると ACK ビットで応答するという動作を繰り返します。この場合、マスタデバイスはマスタ受信モードで、スレーブデバイスはスレーブ送信モードで動作します。

データの最終バイトの通信が完了すると、マスタデバイスはストップビットを送出して転送を終了します。マスタデバイスが受信モードの場合、最後の ACK ビットの代わりにストップビットを送信します。ストップビットは、SCL ラインを High に保ったまま、SDA ラインを Low から High に移行させる事で示されます。

マスタがバスの制御を保持したまま、次の転送を開始する場合があります。そのような場合、マスタデバイスはストップビット (受信モード中ならば最後の ACK ビット) の代わりに、もう一度スタートビットを送出します。

I²Cバスでは、以下の3つのメッセージプロトコルを使います。

- マスタが1つのスレーブにデータを書き込むシングルメッセージ
- マスタが1つのスレーブからデータを読み出すシングルメッセージ
- マスタが1つまたは複数のスレーブに対して、複数回の読み出しまたは書き込み、あるいは読み出しと書き込みの組み合わせを実行するコンバインドメッセージ

あるデバイスが論理 1 を送信 (ラインをフローティングに保持) 中に、もう 1 つのデバイスが論理 0 を送信 (ラインを Low に保持) した場合、前者のデバイスはラインが論理 1 ではない事を検出できます。この検出を SCL ラインに対して適用した場合を、クロックストレッチと呼びます。クロックストレッチを使うと、スレーブデバイスからデータフローを制御できます。上記の検出を SDA ラインに対して適用した場合を、調停と呼びます。調停は、複数のマスタデバイスが同時に通信を行わないようにする仕組みです。

32.3.1 クロックストレッチ

スレーブデバイスがデータ処理を完了していない場合、クロックストレッチで後続データの転送を遅延させる事ができます。スレーブデバイスがビット送受信後に SCL クロックラインを Low に保持する事で、次の通信準備が整っていない事を知らせる事ができます。このスレーブと通信中のマスタは次のビットを転送するために SCL ラインを High にしようとしませんが、クロックラインが解放されていない事を検出します。SCL 接続はオープンドレインであるため、スレーブは通信を再開する準備が整うまでラインを Low に保てます。このように、クロックストレッチを使うとトランスミッタの動作に追いつけない場合にレシーバ側から受信データのフローを制御できます。

32.3.2 調停

各マスタデバイスは、バス上のスタート/ストップビットを監視する必要があります。デバイスはバスがビジー状態である事を検出した場合、バスがアイドル状態に戻るまで新たなメッセージを開始できません。

しかし、2 つのマスタデバイスがほぼ同時に送信を開始しようとする事は考えられます。この時、調停プロセスが開始します。各トランスミッタは SDA データラインのレベルを確認し、予測しているレベルと比較します。これら 2 つのレベルが一致しない事を検出したトランスミッタは調停に敗れ、SDA ラインへの送信を停止します。

例えば、あるトランスミッタが SDA ラインを論理 1 に保持 (フローティングにする) し、もう 1 つのトランスミッタが論理 0 に保持 (Low 駆動) すると、その結果、SDA ラインは Low です。すると、前者のトランスミッタは予測するレベルとラインのレベルが異なる事を検出し、他のトランスミッタが通信中であると判断します。

この差異を検出した前者のトランスミッタは調停に敗れ、SDA ラインの駆動を停止します。このトランスミッタがマスタデバイスでもあった場合、SCL ラインの駆動も停止する必要があります。その後、このトランスミッタはラインを監視し、ストップ条件を検出すると再送を試みます。その間、SDA ラインの予測レベルと実際のレベルに差異を検出しなかったもう一方のデバイスは、伝送を継続します。こちらのデバイスでは特別な処理は不要です。こちらのデバイスには他のトランスミッタによる通信への干渉はなく、予測通りに送信が実行されているように見えるからです。

1 つのマスタが複数のスレーブのアドレスを指定すると、スレーブ送信モードでも調停が実行されますが、それほど一般的ではありません。

2 つのマスタデバイスが 2 つの異なるスレーブデバイスにメッセージを送信している場合、送信しているスレーブアドレスの値が小さい方のマスタが常にアドレスステージでの調停に勝ちます。2 つのマスタデバイスが同じスレーブアドレスにメッセージを送信し、このアドレスが複数のスレーブを指している場合があります。このような場合はデータステージで調停が実行されます。

調停が発生するケースは非常にまれですが、マルチマスタ環境を適切にサポートするには必須のプロセスです。

32.4 I²C モードの動作

全ての MSSP I²C 通信は、バイト単位で処理され、MSb から順にシフト出力されます。6 つの SFR と 2 つの割り込みフラグが、PIC[®] マイクロコントローラとユーザ ソフトウェアをモジュールに結び付けます。このモジュールは、2 本のピン (SDA と SCL) を使って他の外部 I²C デバイスと通信します。

32.4.1 バイト形式

I²C の全ての通信は 9 ビットセグメントで実行されます。マスタからスレーブ (またはその逆) へ 1 バイト送信し、受信側が肯定応答ビットを返します。SCL ラインの 8 番目の立ち下がりがエッジの後、送信側デバイスは SDA ピンを入力に切り換え、次のクロックパルスで ACK 値を受信します。

クロック信号 SCL は、マスタデバイスから供給します。データは SCL 信号が Low の間に変化でき、クロックの立ち上がりエッジでサンプリングされます。SCL ラインが High の間の SDA ラインの変化は、後述する特殊なバス条件を意味します。

32.4.2 I²C 用語の定義

I²C では特有の I²C 通信用語を使います。これらは後出の表で説明し、それ以後は特に説明せずに使います。この表は、Philips 社の I²C 仕様書より引用したものです。

32.4.3 SDA ピンと SCL ピン

I²C モードで SSPEN ビットをセットすると、SCL ピンと SDA ピンはオープンドレインとなります。これらのピンは、対応する TRIS ビットをセットして入力として設定する必要があります。

Note 1: I²C モードが有効の場合、データ出力は 0 になります。

2: PPS モジュールを使うと、SDA および SCL 機能を全てのデバイスピンに割り当てられます。これらの機能は双方向です。SDA 入力には SSPDATPPS レジスタで選択します。SCL 入力には SSPCLKPPS レジスタで選択します。出力は RxyPPS レジスタで選択します。各機能の入出力が同じピンとなるように注意します。

32.4.4 SDA ホールド時間

SDA ピンのホールド時間は、SSPxCON3 レジスタの SDAHT ビットで選択します。ホールド時間とは SCL の立ち下がりがエッジ後に SDA が有効に保持される時間です。SDAHT ビットをセットすると最小ホールド時間として 300 ns が選択され、静電容量の大きいバスに対して有効です。

表 32-2: I²C バスの用語

用語	説明
トランスミッタ	データをバスにシフト出力するデバイス
レシーバ	データをバスからシフト入力するデバイス
マスタ	転送を開始し、クロック信号を生成し、転送を終了するデバイス
スレーブ	マスタでアドレス指定されるデバイス
マルチマスタ	データ転送を開始できるデバイスが複数接続されたバス
調停	バスを制御するマスタが同時に複数存在しないようにする手順 (メッセージの破壊を防ぐ)
同期	バス上の複数のデバイスのクロックを同期させる手順
アイドル	バスを制御するマスタが存在せず、SDA と SCL の両ラインが High の状態
アクティブ	1 つまたは複数のマスタデバイスがバスを制御している状態
アドレス指定されたスレーブ	自身のアドレスに一致するアドレス信号を受け取り、マスタからアクティブにクロック駆動されているスレーブデバイス
一致アドレス	スレーブが受け取り、SSPxADD に格納された値と一致するアドレスバイト
書き込み要求	R/W ビットがクリアされた一致アドレスをスレーブに送り、スレーブにデータ受信の準備を整えさせる事
読み出し要求	R/W ビットをセットしたアドレスバイトをマスタが送出し、スレーブからのデータ送信を要求する事 (データとは、次のバイト以降、反復スタートまたはストップ条件までの全てのバイトを指す)
クロックストレッチ	通信を一時停止するために、バス上のデバイスが SCL を Low に保持する事
バスコリジョン	モジュールが出力中であり SDA ラインが High 状態であると想定しているにも関わらず、同ラインで Low が検出される事

32.4.5 スタート条件

I²C の仕様書は、スタート条件を「SCL ラインが High の時に SDA が High から Low へ遷移する事」と定義しています。スタート条件は常にマスタによって生成され、バスがアイドル状態からアクティブ状態に遷移した事を意味します。図 32-12 に、スタート条件とストップ条件の波形を示します。

モジュールが SDA ラインを Low にアサートする前に Low をサンプリングした場合、スタート条件でバスコリジョンが発生する可能性があります。これは、スタート時にバスコリジョンは発生しないという I²C の仕様準拠していません。

32.4.6 ストップ条件

ストップ条件とは SCL ラインが High の時に SDA ラインが Low から High へ遷移する事です。

Note: ストップ条件が有効になるには SCL に少なくとも 1 回は Low 期間が必要です。従って、SCL ラインが High のまま SDA ラインが High から Low へ遷移し、再び High へ遷移した場合はスタート条件のみが検出されます。

32.4.7 反復スタート条件

反復スタート条件が成立するのは、ストップ条件が成立する場合と同じです。マスタは、転送終了後もバスを解放せずに反復スタート条件を発行できます。反復スタートはスレーブにとってスタートと同じであり、全てのスレーブロジックをリセットしてアドレスを入力できるよう準備をします。マスタは、同じスレーブまたは別のスレーブのアドレスを指定できます。図 32-13 に、反復スタート条件の波形を示します。

10 ビットアドレス指定のスレーブモードの場合、アドレス指定したスレーブからマスタがデータを読み出すために反復スタートが必要です。スレーブの全アドレスが指定され、上位/下位アドレスバイトが両方も一致すると、マスタは反復スタートを発行した後、R/W ビットがセットされた上位アドレスバイトを送ります。そして、スレーブロジックがクロックをホールドしてデータ出力の準備をします。

10 ビットモードの R/W クリアによる完全一致後は、ストップ条件、R/W クリアによる上位アドレス、上位アドレス不一致のいずれかが発生するまで、一致フラグはセットされたまま維持されます。

32.4.8 スタート/ストップ条件の割り込みマスク

SSPxCON3 レジスタの SCIE および PCIE ビットを使うと、通常はサポートされないスレーブモードの割り込み生成を有効にできます。スタート/ストップ条件検出による割り込み機能が既に有効に設定されているスレーブモードでは、これらのビットは影響を与えません。

図 32-12: I²C のスタートおよびストップ条件

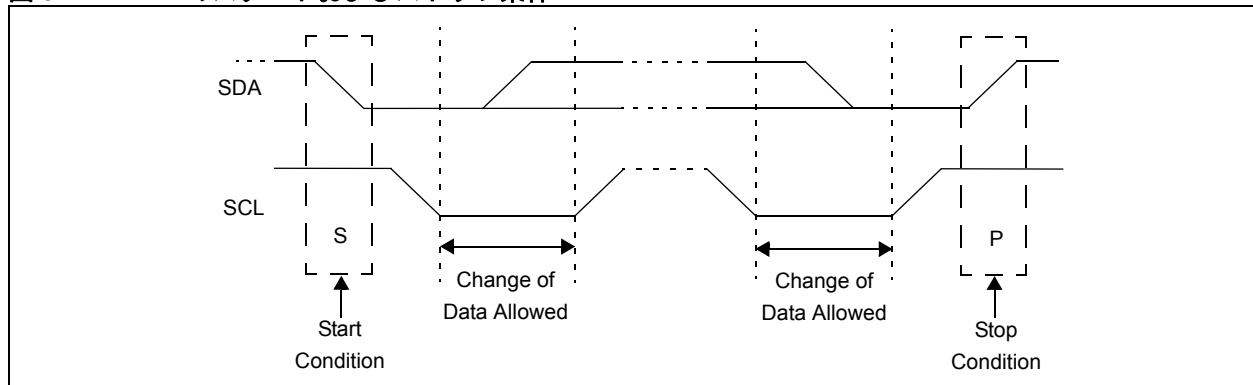
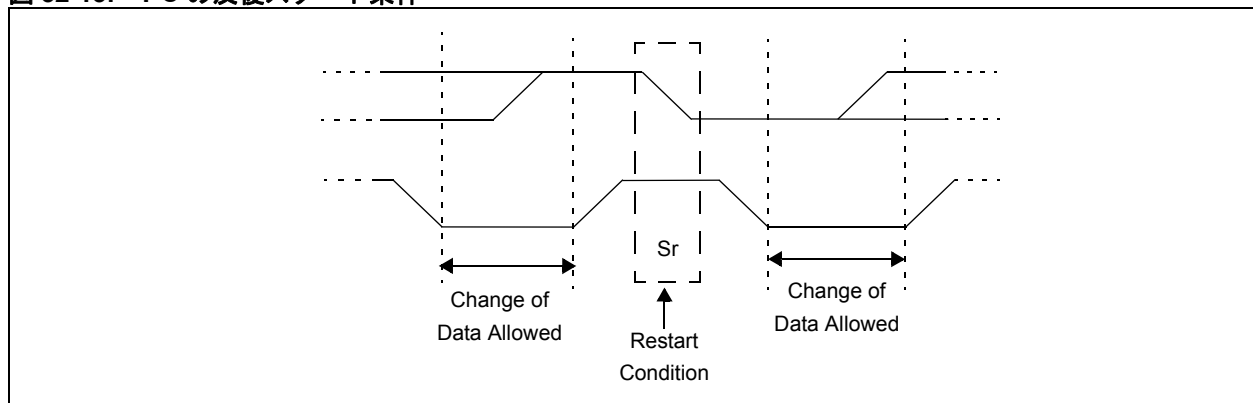


図 32-13: I²C の反復スタート条件



32.4.9 ACK シーケンス

I²C で転送されるバイトの 9 番目の SCL パルスは肯定応答 (ACK) 専用です。このパルスで受信側デバイスは SDA ラインを Low に駆動してトランスミッタに応答します。この間、トランスミッタは応答をシフト入力するためにラインの制御を解放する必要があります。ACK はアクティブ Low 信号であり、SDA ラインを Low に駆動する事で、送信されたデータ受信が完了し、さらに後続のデータを受信する準備が整っている事をトランスミッタに伝えます。

ACK の結果は、SSPxCON2 レジスタの ACKSTAT ビットに格納されます。

AHEN ビットと DHEN ビットがセットされている場合、トランスミッタへ送り返される ACK の値はユーザがスレーブ ソフトウェアで設定できます。SSPxCON2 レジスタの ACKDT ビットをセット / クリアして、応答を決定します。

SSPxCON3 レジスタの AHEN ビットと DHEN ビットがクリアされている場合、スレーブ ハードウェアが ACK 応答を生成します。

スレーブが ACK を生成しない場合があります。それは、バイト受信時に SSPxSTAT レジスタの BF ビットまたは SSPxCON1 レジスタの SSPOV ビットがセットされている場合です。

バス上の SCL の 8 番目の立ち下がりがエッジ後に、このモジュールのアドレスが指定されると、SSPxCON3 レジスタの ACKTIM ビットがセットされます。ACKTIM ビットはアクティブなバスの ACK のタイミングを示します。ACKTIM ステータスビットがアクティブになるのは、AHEN ビットまたは DHEN ビットがイネーブルの時のみです。

32.5 I²C スレーブモードの動作

MSSP スレーブモードには 4 種類の動作モードがあり、SSPxCON1 レジスタの SSPM ビットで選択します。これらのモードは、7 ビットと 10 ビットのアドレス指定モードに分けられます。10 ビットアドレス指定モードは、7 ビットの場合と同じ動作ですが、増加分のアドレスビットを処理する時間 (オーバーヘッド) が余分に必要です。

各アドレス指定モードには、スタートおよびストップ ビット割り込みを有効にするモードもあり、この場合、スタート、反復スタート、ストップ条件を検出すると SSPxIF ビットがセットされます。

32.5.1 スレーブモードアドレス

SSPxADD レジスタ (レジスタ 32-6) はスレーブモードアドレスを格納します。スタート / 反復スタート条件後、最初に受信したバイトとこのレジスタ値が比較されます。バイトが一致すると、その値が SSPxBUF レジスタに書き込まれ、割り込みが生成されます。値が一致しない場合、モジュールはアイドル状態に移行し、ソフトウェアへの通知は何もありません。

SSP マスクレジスタ (レジスタ 32-5) は、アドレス一致プロセスに影響を与えます。詳細は [セクション 32.5.8 「SSP マスクレジスタ」](#) を参照してください。

32.5.1.1 I²C スレーブの 7 ビットアドレス指定モード

7 ビットアドレス指定モードでは、アドレスの一致を判断する際に受信データバイトの LSB が無視されます。

32.5.1.2 I²C スレーブの 10 ビットアドレス指定モード

10 ビットアドレス指定モードの場合、最初の受信バイトがバイナリ値「11110A9A80」と比較されます。A9 と A8 は 10 ビットアドレスの上位 2 ビットで、SSPxADD レジスタの bit 2 と bit 1 に格納されています。

上位バイトの肯定応答 (ACK) 後、ユーザが SSPxADD レジスタに下位アドレスを書き込むまで UA ビットがセットされ SCL が Low に保持されます。下位アドレスバイトが入力されると、8 ビット全てが SSPxADD レジスタにある下位アドレスの値と比較されます。アドレスが一致しない場合でも、SSPxIF と UA がセットされ、SSPxADD が更新されて再び上位バイトを受信するまで SCL は Low に保持されます。SSPxADD が更新されると、UA ビットはクリアされます。これにより、モジュールは次の通信で上位アドレスバイトを確実に受信する準備を整える事ができます。

全ての 10 ビットアドレス指定通信開始時には、書き込み要求として上位 / 下位アドレスが一致する必要があります。スレーブをアドレス指定したら反復スタートを発行し、R/W ビットをセットした上位アドレスを送出する事で通信を開始できます。そしてスレーブ ハードウェアが読み出し要求に ACK を返し、データ送信の準備をします。この動作は、スレーブが上位 / 下位アドレスバイト一致を受信した後にのみ有効です。

32.5.2 スレーブ受信

受信したアドレスが一致し、その $\overline{R/W}$ ビットがクリアされていた場合、SSPxSTAT レジスタの $\overline{R/W}$ ビットはクリアされます。受信したアドレスは SSPxBUF レジスタに転送され、肯定応答 (ACK) が返されます。

受信アドレスがオーバーフロー条件を満たしている場合、否定応答 (NACK) が生成されます。オーバーフロー条件とは、SSPxSTAT レジスタの BF ビットまたは SSPxCON1 レジスタの SSPOV ビットがセットされている事と定義されます。この動作を変更するには、SSPxCON3 レジスタの BOEN ビットを使います。詳細は [レジスタ 32-4](#) を参照してください。

1 データバイトが転送されるたびに MSSP 割り込みが生成されます。フラグビット (SSPxIF) は、ソフトウェアでクリアする必要があります。

SSPxCON2 レジスタの SEN ビットをセットすると、各バイトの受信後に SCLx は LOW に保持されます (クロック ストレッチ)。この場合、SSPxCON1 レジスタの CKP ビットをセットしてクロックを解放する必要があります (10 ビットモードの場合は例外あり)。詳細は [セクション 32.5.6.2 「10 ビットアドレス指定モード」](#) を参照してください。

32.5.2.1 7 ビットアドレス指定の受信

このセクションでは、7 ビットアドレス指定モードの I²C スレーブとして設定された MSSP モジュールの一般的なイベント シーケンスについて説明します。[図 32-14](#) と [図 32-15](#) に、このシーケンスの波形を示します。

以下に、I²C 通信の代表的なステップを示します。

1. スタートビットを検出する。
2. SSPxSTAT の S ビットがセットされる。スタート検出割り込みが有効の場合は SSPxIF ビットもセットされる。
3. $\overline{R/W}$ ビットがクリアされた一致アドレスをスレーブが受信する。
4. スレーブが SDA を Low に駆動してマスタに \overline{ACK} を送り、SSPxIF ビットをセットする。
5. ソフトウェアは SSPxIF ビットをクリアする。
6. ソフトウェアは SSPxBUF から受信アドレスを読み出し、BF フラグがクリアされる。
7. SEN=1 の場合、スレーブ ソフトウェアが CKP ビットをセットして SCL ラインを解放する。
8. マスタがデータバイトを送信する。
9. スレーブが SDA を Low に駆動してマスタに \overline{ACK} を送信し、SSPxIF ビットをセットする。
10. ソフトウェアは SSPxIF をクリアする。
11. ソフトウェアは SSPxBUF から受信バイトを読み出し、BF がクリアされる。
12. マスタからの全ての受信バイトに対して、ステップ 8 ~ 12 を繰り返す。
13. マスタが SSPxSTAT の P ビットをセットしてストップ条件を送信し、バスがアイドル状態に移行する。

32.5.2.2 AHEN と DHEN を使う 7 ビット受信

スレーブデバイスの受信で AHEN と DHEN をセットした場合、受信動作はこれらの設定をしない場合と同じですが、SCL の 8 番目の立ち下がりエッジの後に割り込みとクロック ストレッチが追加されます。これらの割り込みによって、ハードウェアではなくスレーブソフトウェアが、受信アドレスバイトまたは受信データバイトに対して \overline{ACK} を生成するかどうかを判断できます。この機能により、以前はこのモジュールでサポートしていなかった PMBus™ をサポートします。

I²C 通信でこれらのオプションを使う場合にスレーブソフトウェアが実行する必要がある手順を以下に示します。[図 32-16](#) に、アドレスホールドとデータホールドの両方を使う場合のモジュール動作を示します。[図 32-17](#) に、SSPxCON2 レジスタの SEN ビットもセットした場合のモジュール動作を示します。

1. SSPxSTAT の S ビットがセットされる。スタート検出割り込みが有効の場合は SSPxIF ビットもセットされる。
2. $\overline{R/W}$ ビットがクリアの状態で一一致アドレスが取り込まれる。SSPxIF がセットされ、SCL の 8 番目の立ち下がりエッジの後に CKP がクリアされる。
3. スレーブは SSPxIF をクリアする。
4. スレーブは、SSPxCON3 レジスタの ACKTIM ビットを確認する事で、SSPxIF ビットが \overline{ACK} の前後いずれであるかを判断できる。
5. スレーブは SSPxBUF からアドレス値を読み出し、BF フラグがクリアされる。
6. スレーブは ACKDT をセットして、 \overline{ACK} 値をマスタに送出する。
7. スレーブが CKP ビットをセットしてクロックを解放する。
8. \overline{ACK} の後に SSPxIF がセットされる (NACK の場合はセットされない)。
9. SEN=1 の場合、スレーブ ハードウェアは \overline{ACK} の後にクロックをストレッチする。
10. スレーブは SSPxIF をクリアする。

Note: クロック ストレッチが実行されず、BF フラグがクリアされた場合でも、SSPxIF ビットは SCL の 9 番目の立ち下がりエッジ後にセットされます。NACK がマスタへ送られた場合のみ、SSPxIF はセットされません。

11. 受信データバイトについて SCL の 8 番目の立ち下がりエッジ後に SSPxIF ビットがセットされて CKP ビットがクリアされる。
12. スレーブが SSPxCON3 レジスタの ACKTIM ビットを読み出して割り込み要因を判定する。
13. スレーブは SSPxBUF から受信データを読み出し、BF がクリアされる。
14. 1 データバイトを受信するたびに、ステップ 7 ~ 14 を繰り返す。
15. スレーブが $\overline{ACK}=1$ を送るか、マスタがストップ条件を送ると、通信が終了する (ストップ検出割り込みが無効に設定されている状態でストップ条件が送信された場合、スレーブが通信終了を認識する唯一の方法は SSP1STAT レジスタの P ビットをポーリングする事です)。

図 32-14: I²C スレーブ、7 ビットアドレス、受信 (SEN=0、AHEN=0、DHEN=0)

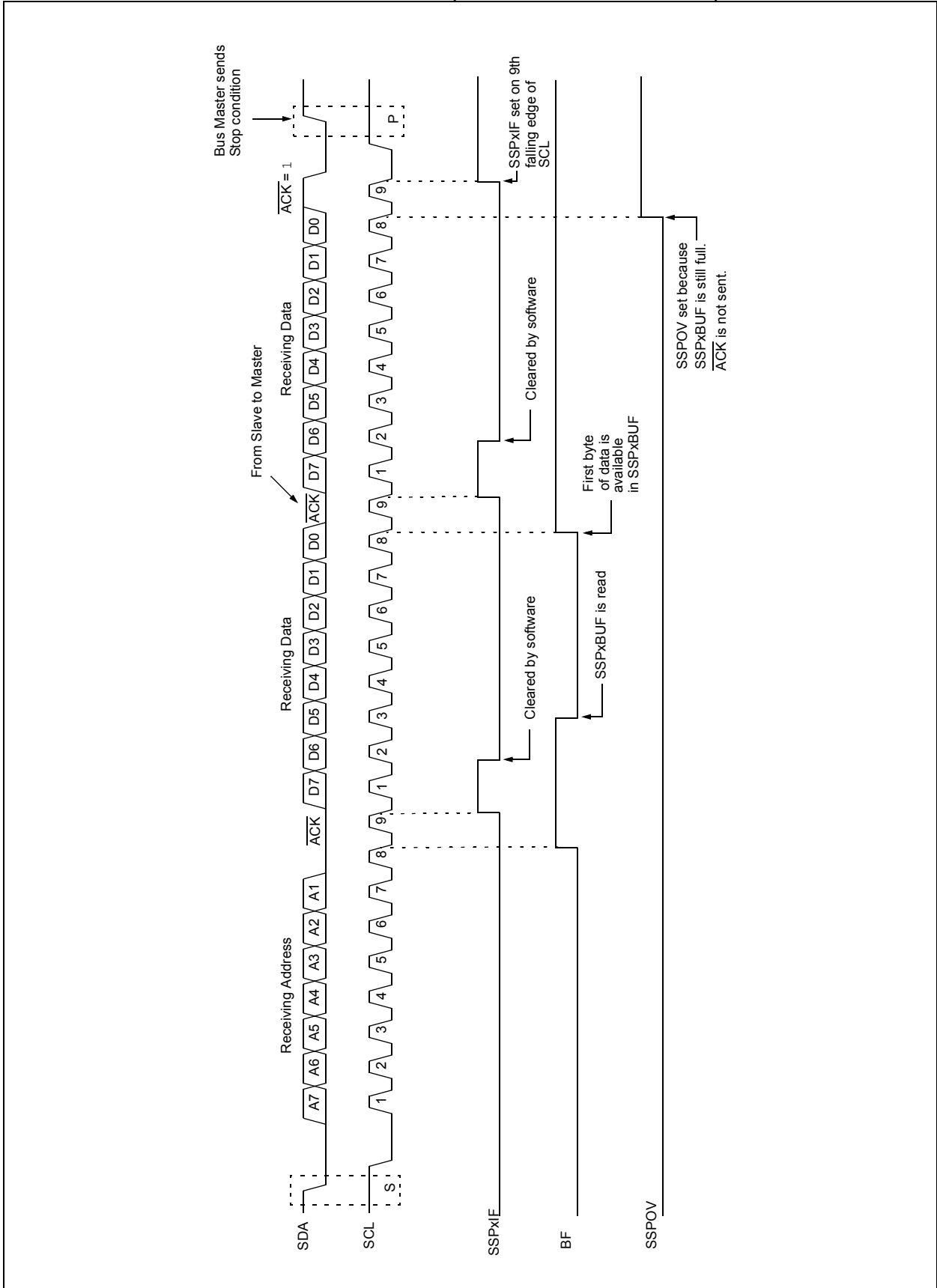
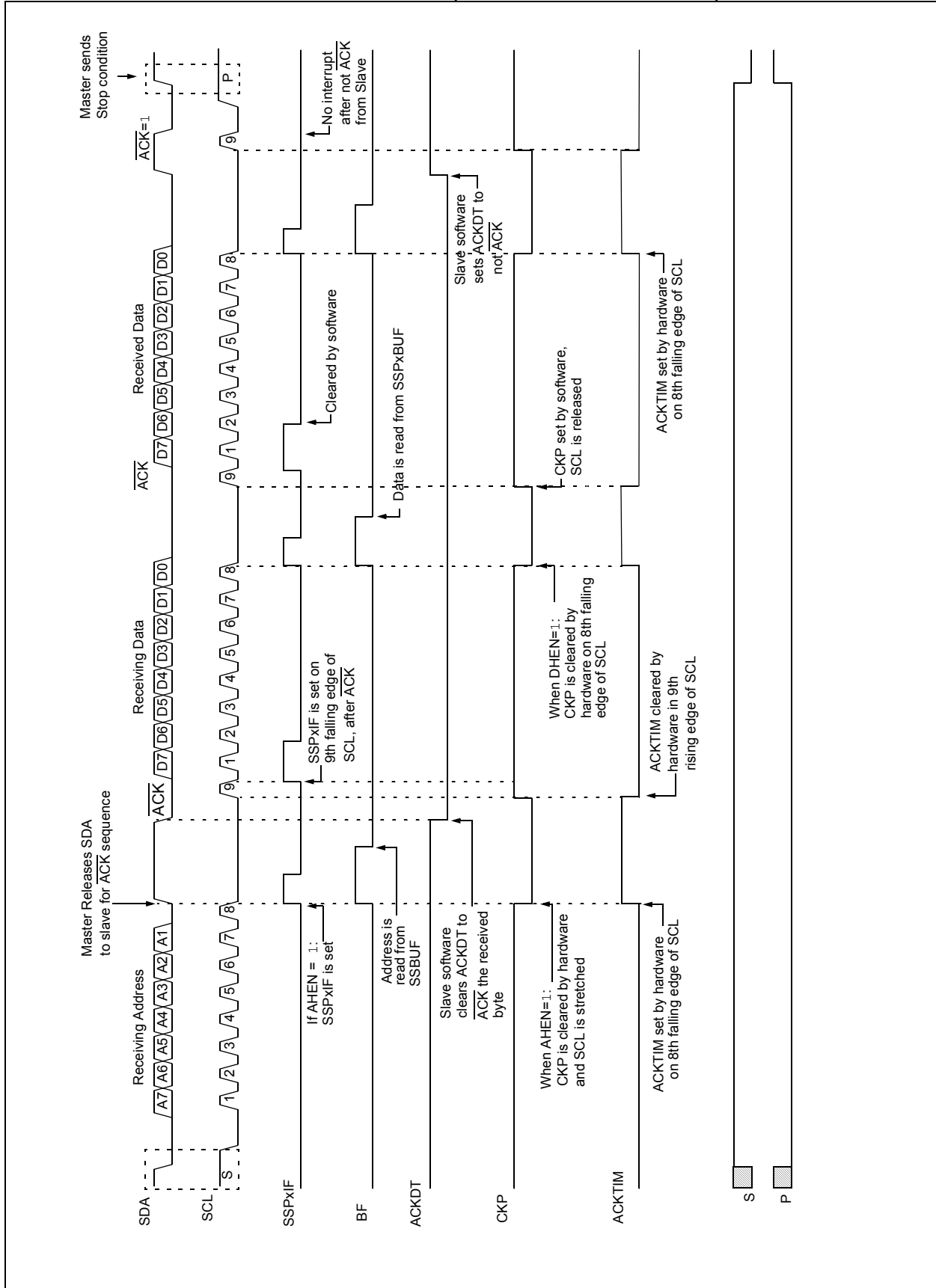
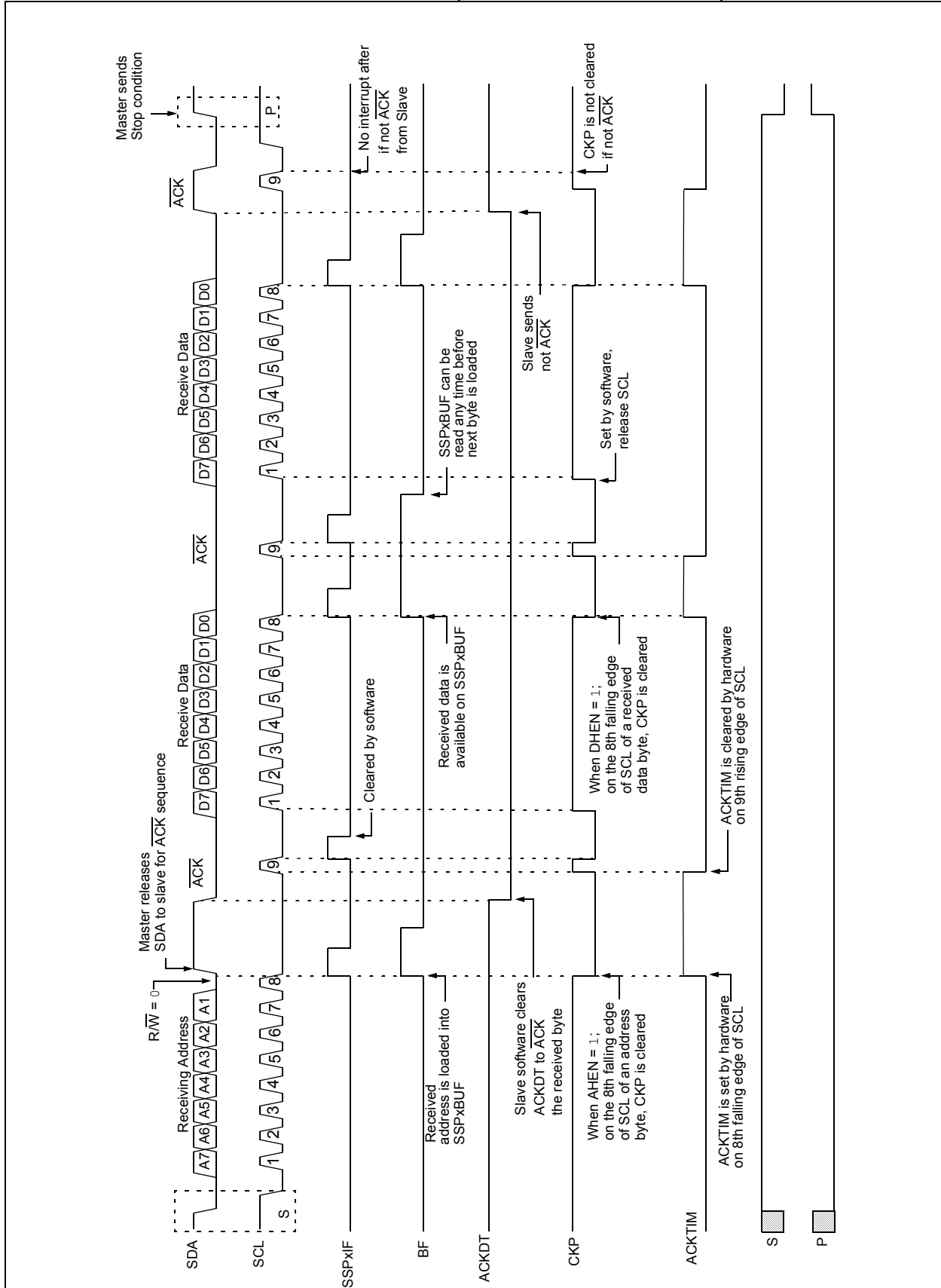


図 32-16: I²C スレーブ、7 ビットアドレス、受信 (SEN=0、AHEN=1、DHEN=1)



PIC16(L)F1764/5/8/9

図 32-17: I²C スレーブ、7 ビットアドレス、受信 (SEN=1、AHEN=1、DHEN=1)



32.5.3 スレーブ送信

R/W ビットがセットされた一致アドレスを受け取ると、SSPxSTAT レジスタの R/W ビットがセットされます。受け取ったアドレスが SSPxBUF レジスタに書き込まれると、スレーブは 9 番目のビットで ACK パルスを送ります。

ACKの後、スレーブハードウェアがCKPビットをクリアして SCL ピンが Low に保持されます (**セクション 32.5.6「クロックストレッチ」** 参照)。クロックストレッチを実行すると、スレーブの送信データ準備が完了するまでマスタは次のクロックパルスをアサートできなくなります。

送信データは SSPxBUF レジスタに書き込まれます。データはさらに SSPSR レジスタにも書き込まれます。その後、SSPxCON1 レジスタの CKP ビットをセットする事で SCL ピンが解放されます。8 ビットのデータが SCL 入力の立ち下がりエッジでシフト出力されます。これにより、SCL が High の間は SDA 信号が確実に有効になります。

マスタレシーバからの ACK パルスが 9 番目の SCL 入力パルスの立ち上がりエッジでラッチされます。この ACK 値は、SSPxCON2 レジスタの ACKSTAT ビットへコピーされます。ACKSTAT がセット (NOT ACK) されると、データ転送は完了です。この場合、NOT ACK がスレーブでラッチされると、スレーブがアイドル状態に移行して次のスタートビットが到達するまで待機します。SDA ラインが Low (ACK) の場合、次の送信データを SSPxBUF レジスタに書き込む必要があります。この場合も、CKP ビットをセットして SCL ピンを解放する必要があります。

1 データバイトが転送されるたびに MSSP 割り込みが生成されます。SSPxIF ビットはソフトウェアでクリアする必要があり、バイトのステータスを判断するには SSPxSTAT レジスタを使います。SSPxIF ビットは 9 番目のクロックパルスの立ち下がりエッジでセットされます。

32.5.3.1 スレーブモードのバスコリジョン

スレーブは読み出し要求を受け取ると SDA ラインにデータをシフト出力し始めます。SSPxCON3 レジスタの SBCDE ビットがセットされている場合にバスコリジョンが検出されると、PIR2 レジスタの BCL1IF ビットがセットされます。バスコリジョンが検出されるとスレーブはアイドル状態に移行し、再度アドレス指定されるまで待機します。ユーザ ソフトウェアは BCL1IF ビットによってスレーブバスコリジョンを処理できます。

32.5.3.2 7 ビット送信

マスタデバイスは、スレーブデバイスへ読み出し要求を送信し、スレーブからデータを読み出します。ここでは、通常の送信動作でスレーブソフトウェアが実行すべきステップを説明します。図 32-18 に、この動作の波形を示します。

1. マスタが SDA と SCL にスタート条件を送信する。
2. SSPxSTAT の S ビットがセットされる。スタート検出割り込みが有効の場合は SSPxIF ビットもセットされる。
3. R/W ビットがセットされた一致アドレスをスレーブが受け取る
4. スレーブハードウェアが ACK を生成して SSPxIF ビットをセットする。
5. ユーザによって SSPxIF ビットがクリアされる。
6. ソフトウェアが SSPxBUF から受信アドレスを読み出して、BF をクリアする。
7. R/W がセットされているため、CKP ビットは ACK の後に自動的にクリアされている。
8. スレーブソフトウェアが SSPxBUF に送信データを書き込む。
9. CKP ビットがセットされて SCL が解放されると、マスタはスレーブからデータを読み出す事ができる。
10. マスタからの ACK 応答が ACKSTAT レジスタに書き込まれると SSPxIF ビットがセットされる。
11. SSPxIF ビットがクリアされる。
12. スレーブソフトウェアは、ACKSTAT ビットをチェックしてマスタに後続の送信データがあるかどうか確認する。

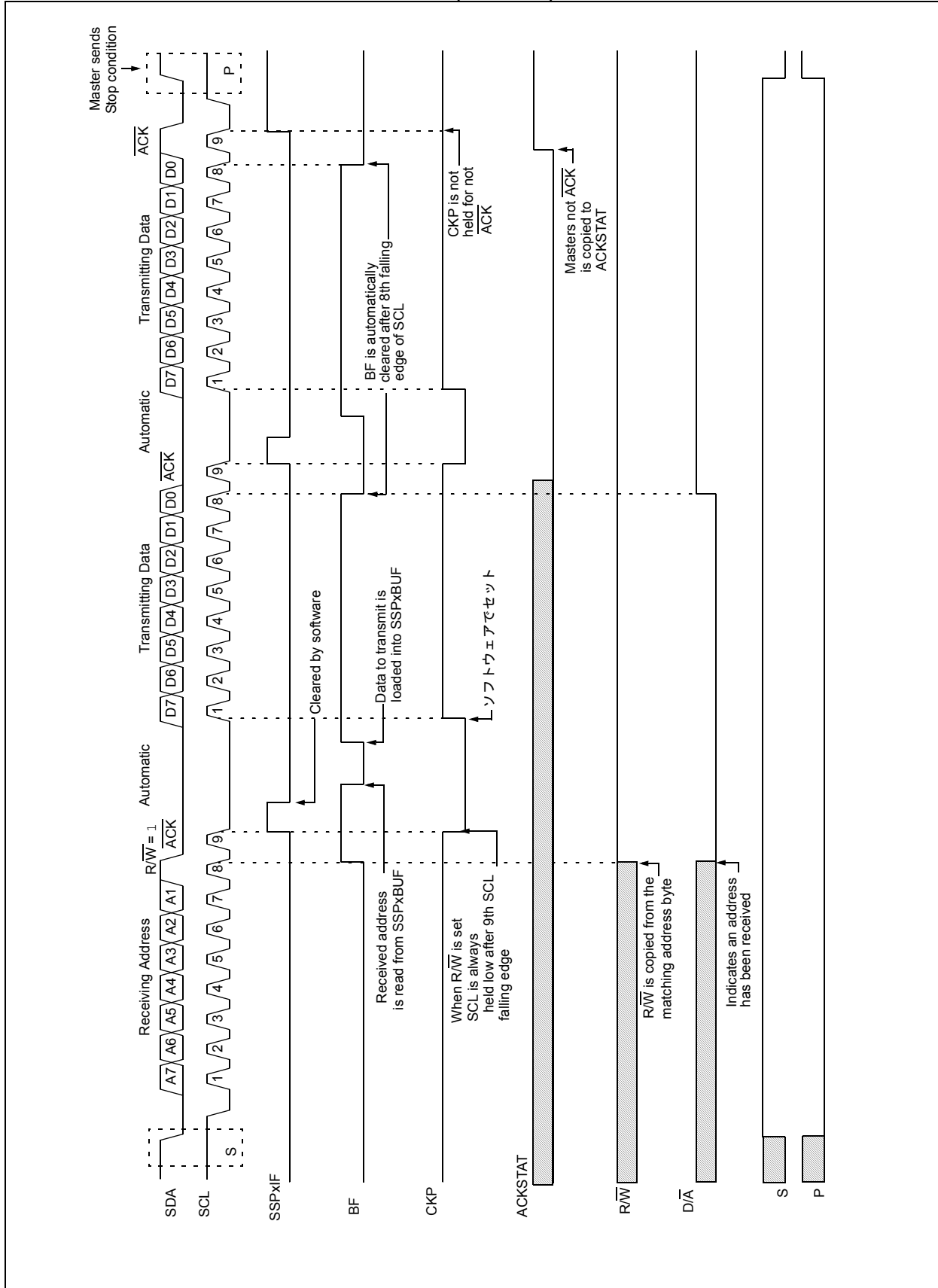
Note 1: マスタが ACK を生成すると、クロックがストレッチされます。

2: ACKSTAT は、SCL の 9 番目の立ち上がりエッジ (立ち下がりエッジではない) で更新される唯一のビットです。

13. 1 バイトを送信するたびに、ステップ 9 ~ 13 を繰り返す。
14. マスタが NOT ACK を送信した場合、クロックはホールドされませんが、SSPxIF ビットはセットされる。
15. マスタが反復スタート条件またはストップ条件を送信する。
16. スレーブのアドレス指定が解除される。

PIC16(L)F1764/5/8/9

図 32-18: I²C スレーブ、7 ビットアドレス、送信 (AHEN = 0)



32.5.3.3 7ビット送信 (アドレスホールド機能が有効)

SSPxCON3 レジスタのAHENビットをセットすると、受け取った一致アドレスの8番目の立ち下がりエッジ後に、さらにクロックストレッチと割り込み生成を追加できます。一致アドレスの入力が完了すると、CKPがクリアされて SSPxIF 割り込みがセットされます。

図 32-19 に、AHEN を有効にした7ビットアドレスのスレーブ送信の一般的な波形を示します。

1. バスはアイドル状態にある。
2. マスタがスタート条件を送る。SSPxSTAT の S ビットがセットされる。スタート検出割り込みが有効の場合は SSPxIF がセットされる。
3. マスタが、 $\overline{R/W}$ ビットをセットした一致アドレスを送る。SCL ラインの8番目の立ち下がりエッジの後、CKP ビットがクリアされて SSPxIF 割り込みが生成される。
4. スレーブ ソフトウェアは SSPxIF をクリアする。
5. スレーブ ソフトウェアが SSPxCON3 レジスタの ACKTIM ビットと、SSPxSTAT レジスタの $\overline{R/W}$ と D/A を読み出して、割り込み要因を決定する。
6. スレーブは SSPxBUF レジスタのアドレス値を読み出し、BF ビットがクリアされる。
7. スレーブ ソフトウェアは、この情報に基づいて \overline{ACK} または NOT ACK のどちらかを判断し、それに応じて SSPxCON2 レジスタの ACKDT ビットをセットする。
8. スレーブが CKP ビットをセットして SCL を解放する。
9. マスタはスレーブが送った \overline{ACK} 値を受け取る。
10. $\overline{R/W}$ ビットがセットされていた場合、スレーブ ハードウェアは ACK の後自動的に CKP ビットをクリアし、SSPxIF ビットをセットする。
11. スレーブ ソフトウェアは SSPxIF をクリアする。
12. スレーブが、マスタへ送信する値を SSPxBUF に書き込んで BF ビットをセットする。

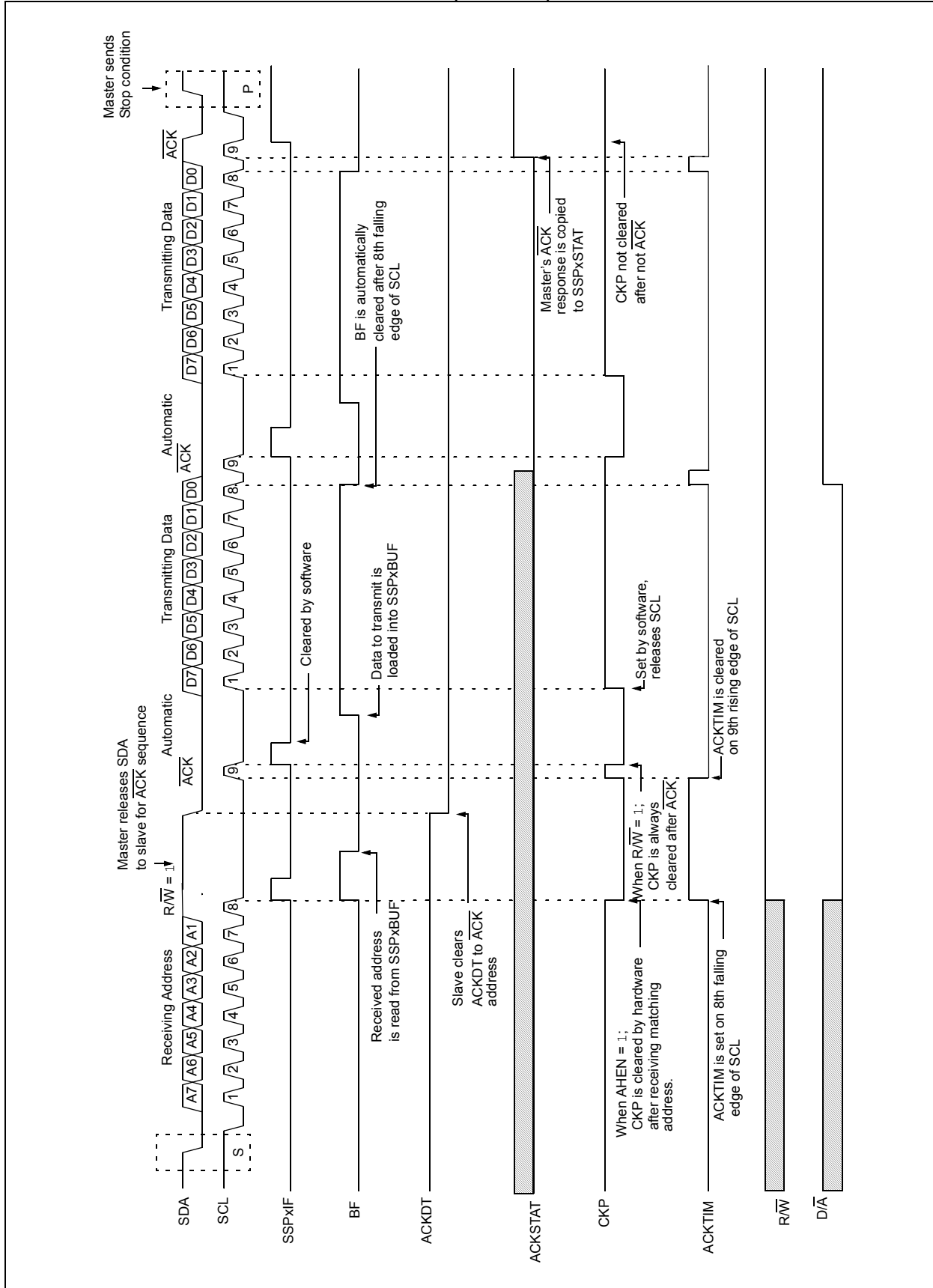
Note: ACK が終わるまで、SSPxBUF にデータを書き込む事はできません。

13. スレーブが CKP ビットをセットしてクロックを解放する。
14. マスタはスレーブからデータを受信し、SCL の9番目のパルスで ACK 値を送る。
15. この \overline{ACK} 値を、スレーブ ハードウェアが SSPxCON2 レジスタの ACKSTAT ビットへコピーする。
16. スレーブからマスタへ 1 バイトを送信するたびに、ステップ 10 ~ 15 を繰り返す。
17. マスタが NOT \overline{ACK} を送るとスレーブがバスを解放するため、マスタはストップ条件を送って通信を終了できる。

Note: スレーブが確実に SCL ラインを解放してストップ条件を受け取る事ができるように、マスタは最後のバイトで NOT ACK を送る必要があります。

PIC16(L)F1764/5/8/9

図 32-19: I²C スレーブ、7 ビットアドレス、送信 (AHEN = 1)



32.5.4 スレーブモードの 10 ビットアドレス 受信

このセクションでは、10 ビットアドレス指定モードの I²C スレーブとして設定された MSSP モジュールの一般的なイベント シーケンスについて説明します。

図 32-20 に、このシーケンスの波形を示します。

以下に、I²C 通信でスレーブ ソフトウェアが実行すべき一般的なステップを示します。

1. バスはアイドル状態にある。
2. マスタがスタート条件を送る。SSPxSTAT の S ビットがセットされる。スタート検出割り込みが有効の場合は SSPxIF がセットされる。
3. マスタが、R/W ビットがクリアされた一致上位アドレスを送る。
4. スレーブは ACK を送り、SSPxIF がセットされる。
5. ソフトウェアは SSPxIF ビットをクリアする。
6. ソフトウェアは SSPxBUF から受信アドレスを読み出し、BF フラグがクリアされる。
7. スレーブが SSPxADD に下位アドレスを書き込み、SCL を解放する。
8. マスタが、一致下位アドレスバイトをスレーブへ送る。UA ビットがセットされる。

Note: ACK シーケンスが完了するまで、SSPxADD レジスタの更新は許可されません。

9. スレーブは ACK を送り、SSPxIF がセットされる。

Note: 下位アドレスが一致しない場合、SSPxIF と UA はセットされたままであるため、スレーブ ソフトウェアは SSPxADD を上位アドレスの値に戻す事ができます。アドレスが一致していないため、BF フラグはセットされません。CKP には影響しません。

10. スレーブは SSPxIF をクリアする。
11. スレーブは受け取った一致アドレスを SSPxBUF から読み出し、BF ビットがクリアされる。
12. スレーブは SSPxADD に上位アドレスを書き込む。
13. マスタがスレーブヘデータバイトを送信し、9 番目の SCL パルスでスレーブの ACK を出力し、SSPxIF ビットがセットされる。
14. SSPxCON2 の SEN ビットがセットされている場合、ハードウェアで CKP ビットがクリアされてクロックがストレッチされる。
15. スレーブは SSPxIF をクリアする。
16. スレーブは SSPxBUF から受信バイトを読み出し、BF がクリアされる。
17. SEN ビットがセットされている場合、スレーブは SCL を解放するために CKP をセットする。
18. 1 バイトを受信するたびに、ステップ 13 ~ 17 を繰り返す。
19. マスタがストップ条件を送り、通信を終了する。

32.5.5 10 ビットアドレス指定 (アドレス / データホールドあり)

AHEN または DHEN をセットした 10 ビットアドレス指定を使う受信動作は、7 ビットモードの場合と同じです。唯一の相違点は、UA ビットを使って SSPxADD レジスタを更新する必要がある事です。CKP ビットがクリアされ、かつ SCL ラインが Low に保持されている場合、全ての機能は 7 ビットモードと同じです。図 32-21 にスレーブ (10 ビットアドレス指定、AHEN = 1) の例を示します。

図 32-22 に、10 ビットアドレス指定モードでのスレーブ トランスミッタの一般的な波形を示します。

PIC16(L)F1764/5/8/9

図 32-20: I²C スレーブ、10 ビットアドレス、受信 (SEN=1、AHEN=0、DHEN=0)

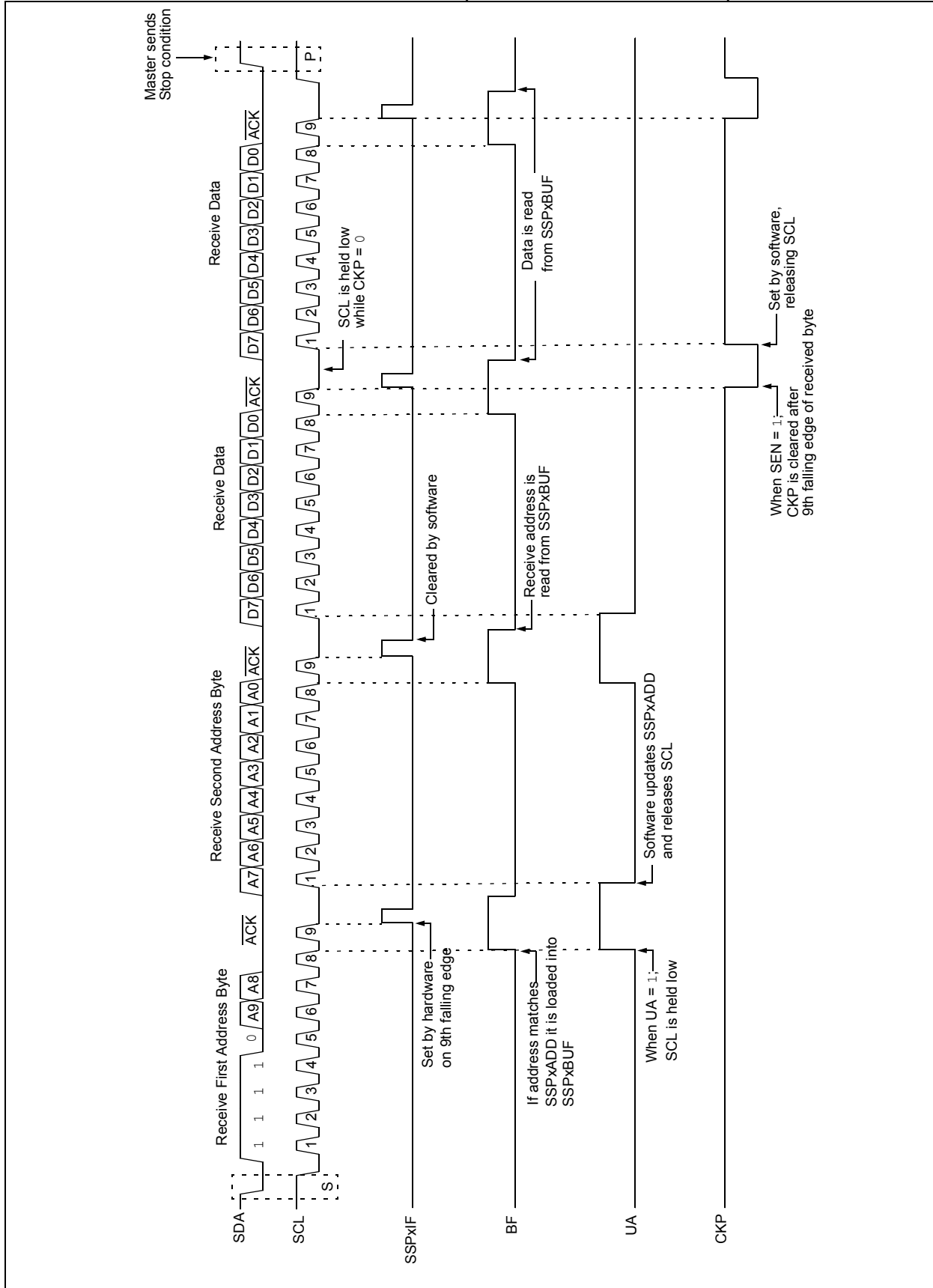
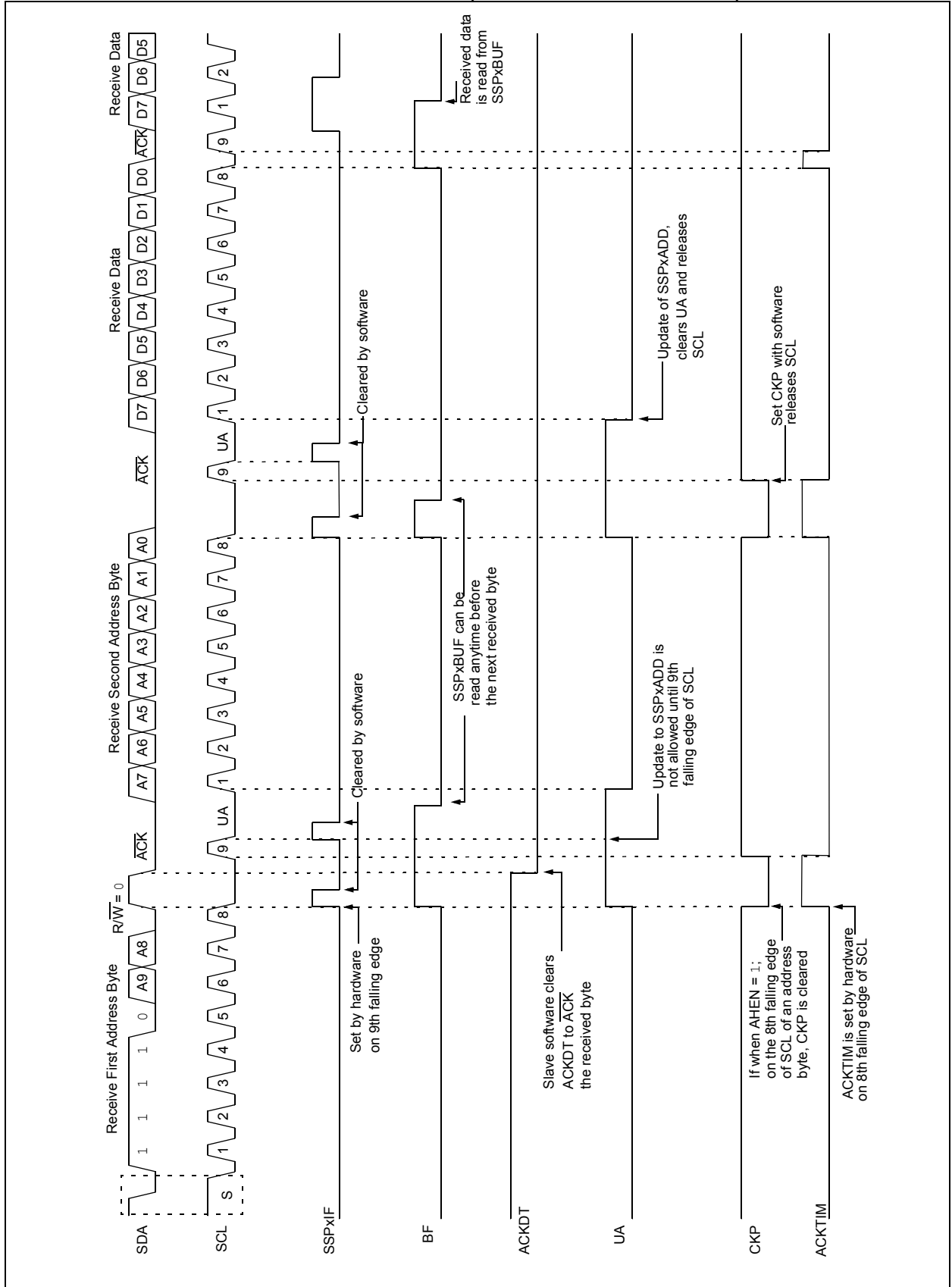
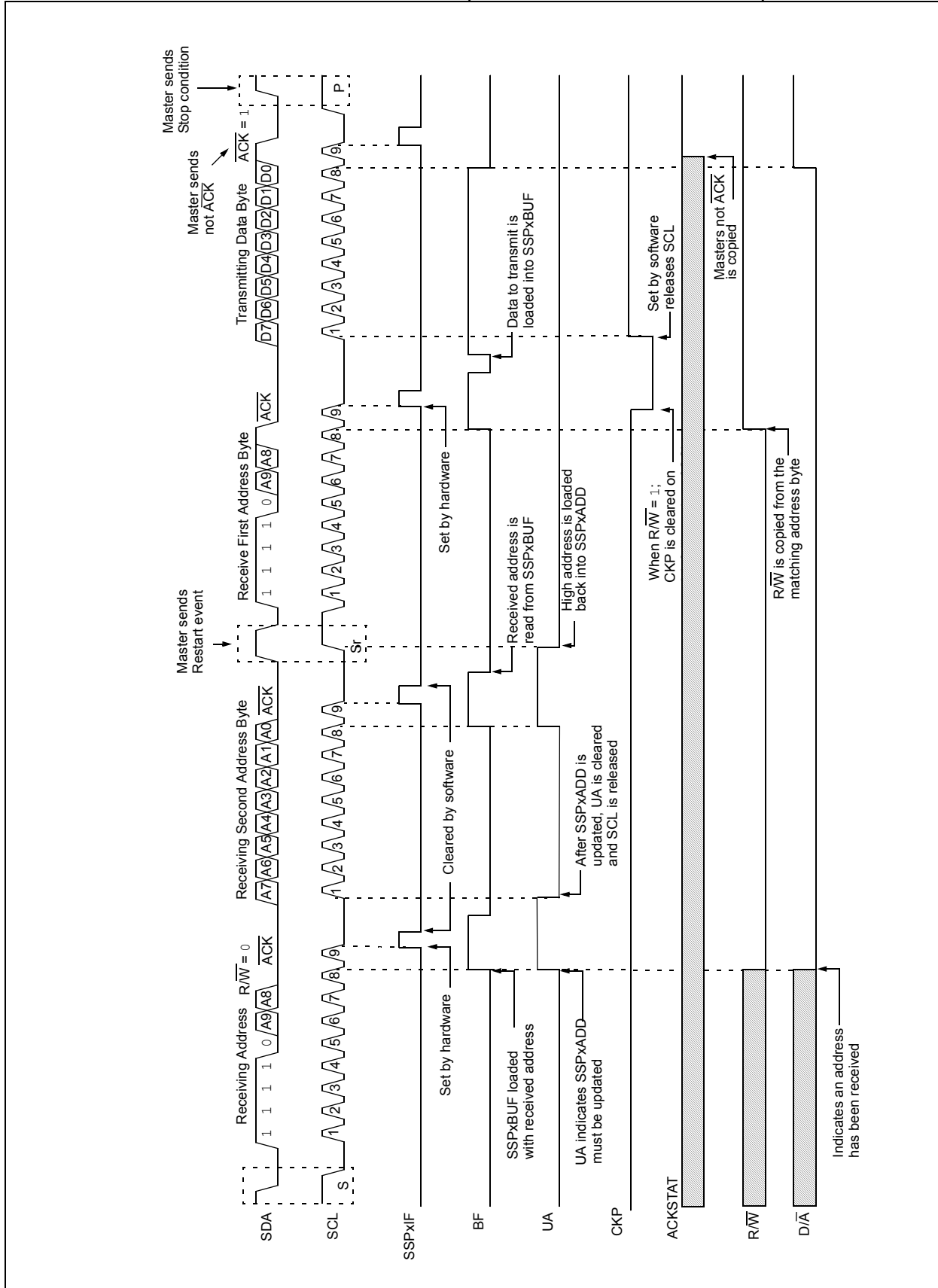


図 32-21: I²C スレーブ、10 ビットアドレス、受信 (SEN=0、AHEN=1、DHEN=0)



PIC16(L)F1764/5/8/9

図 32-22: I²C スレーブ、10 ビットアドレス、送信 (SEN = 0、AHEN = 0、DHEN = 0)



32.5.6 クロック ストレッチ

クロック ストレッチとは、バス上に接続されたデバイスが SCL ラインを Low に保持する事で通信を停止させる事です。スレーブは、このクロック ストレッチ機能を使って、データ処理またはマスタへの応答準備の時間を作る事ができます。マスタは、この機能とは無関係に動作し常にバスでアクティブ状態ですが、ストレッチ中はデータを送信しません。マスタ ソフトウェアは、スレーブによるクロック ストレッチを認識できないため、この動作は SCL を生成するハードウェアが処理します。ソフトウェアによるストレッチ制御には、SSPxCON1 レジスタの CKP ビットを使います。CKP ビットがクリアされると、モジュールは SCL ラインが Low になるまで待機し、その後 Low を保持します。CKP をセットすると、SCL が解放されて通信が再開します。

32.5.6.1 通常のクロック ストレッチ

SSPxSTAT レジスタの \overline{RW} ビットがセットされ、読み出し要求が実行された場合、ACK パルスの後にスレーブハードウェアが CKP をクリアします。これによって、スレーブはマスタへ送信するデータを SSPxBUF に読み込む時間を確保できます。SSPxCON2 レジスタの SEN ビットがセットされている場合、スレーブハードウェアは ACK シーケンスの後に毎回クロック ストレッチを実行します。スレーブの準備が整うと、ソフトウェアで CKP がセットされて通信が再開します。

- Note 1:** BF ビットは、クロック ストレッチを実行するかどうかに影響しません。これは、以前のバージョンのモジュールと異なります。以前は、SCL の 9 番目の立ち下がりエッジより前に SSPxBUF が読み出された場合、クロック ストレッチが実行されず、CKP がクリアされました。
- 2:** 以前のモジュールバージョンでは、SCL の 9 番目の立ち下がりエッジより前に SSPxBUF ヘデータが書き込まれた場合、送信用のクロックはストレッチされませんでした。現バージョンでは、読み出し要求に対して常に CKP ビットがクリアされます。

32.5.6.2 10 ビットアドレス指定モード

10ビットアドレス指定モードの場合、UAビットがセットされるとクロックは常にストレッチされます。これは、CKP がクリアされずに SCL がストレッチされる唯一の状況です。SSPxADD への書き込みが実行されるとすぐに SCL は解放されます。

Note: 以前のバージョンのモジュールでは、2 番目のアドレスバイトが一致しない場合、モジュールはクロックをストレッチしませんでした。

32.5.6.3 バイト単位の NACK

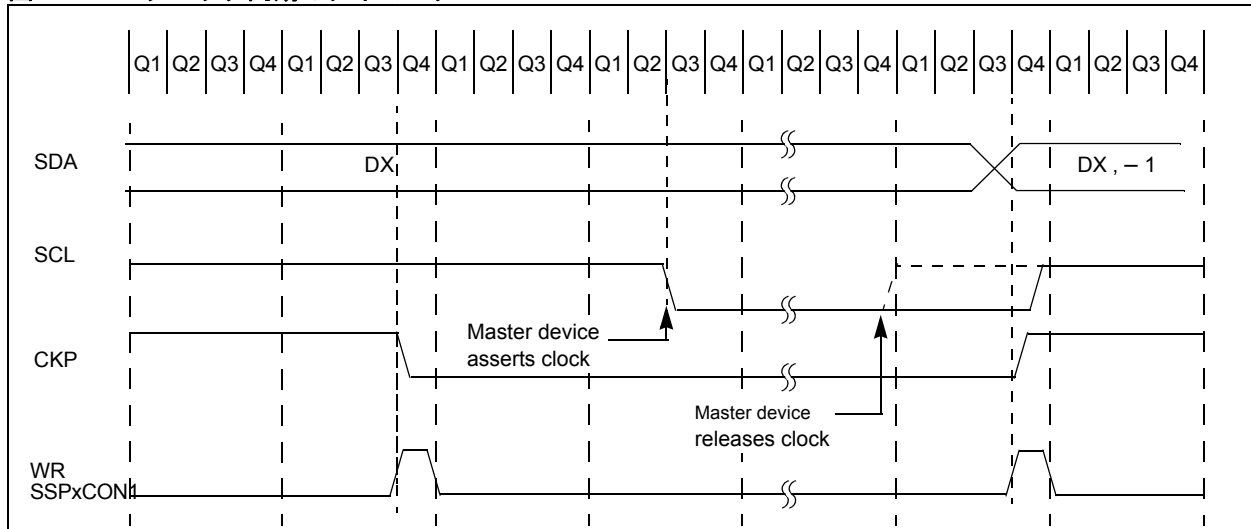
SSPxCON3のAHENビットがセットされている場合、CKP は一致アドレスバイトを受信する SCL の 8 番目の立ち下がりエッジの後にハードウェアによってクリアされます。SSPxCON3 の DHEN ビットがセットされている場合、CKP は受信データに対して SCL の 8 番目の立ち下がりエッジの後にクリアされます。

SCL の 8 番目の立ち下がりエッジの後でクロックがストレッチされると、スレーブは受信アドレスまたは受信データを見て、受信データに ACK を返すかどうかを判断できます。

32.5.6.4 クロック同期と CKP ビット

CKP ビットがクリアされると、モジュールは SCL ラインが Low になるまで待機し、その後 Low を保持します。しかし、CKP ビットをクリアしても、SCL 出力が Low にサンプリングされるまで、SCL 出力の Low はアサートされません。従って、外部 I²C マスタデバイスが既に SCL ラインをアサートしていないと、CKP ビットは SCL ラインをアサートしません。SCL 出力は、CKP ビットがセットされて I²C バス上のその他全てのデバイスが SCL を解放するまで Low を保持します。これにより、CKP ビットへの書き込みによって SCL の High 期間が規格下限を下回る事を防止できます (図 32-23 参照)。

図 32-23: クロック同期のタイミング



PIC16(L)F1764/5/8/9

32.5.7 ジェネラルコールアドレスのサポート

I²C バスのアドレス指定では通常、スタート条件の次のバイトで、マスタデバイスがアドレス指定するスレーブデバイスを示します。例外は、全てのデバイスのアドレスを指定するジェネラルコールアドレスです。このアドレスを使うと、理論上は全デバイスが ACK を返します。

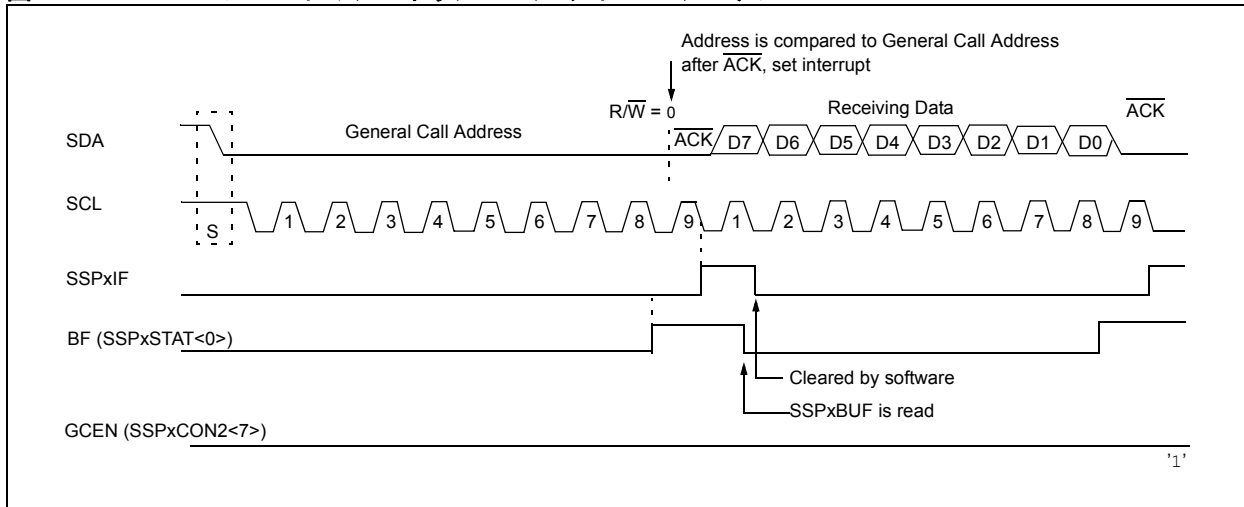
ジェネラルコールアドレスは I²C プロトコルの予約済みアドレスの 1 つで、「0x00」と定義されています。SSPxCON2 レジスタの GCEN ビットがセットされている場合、スレーブ モジュールは SSPxADD の値に関係なく、このアドレス受信に対して自動的に ACK を返します。スレーブが、R/W ビットのクリアされた全ての 0 のアドレス値を受信後に割り込みが生成され、ス

レーブソフトウェアが SSPxBUF を読み出して応答できます。図 32-24 に、ジェネラルコールの受信シーケンスを示します。

10 ビットアドレス指定モードの場合、ジェネラルコールアドレスの受信で UA ビットはセットされません。スレーブは、7 ビットモードと同じように 2 番目のバイトをデータとして受信する準備をします。

SSPxCON3 レジスタの AHEN ビットがセットされている場合、通常アドレス受信と同じように、スレーブハードウェアは SCL の 8 番目の立ち下がりエッジ後にクロックをストレッチします。その後、スレーブは ACKDT ビットの値をセットして、通常の通信同様クロックを解放する必要があります。

図 32-24: スレーブモードのジェネラルコールアドレスシーケンス



32.5.8 SSP マスクレジスタ

I²C スレーブモードでは、SSP マスク (SSPxMSK) レジスタ (レジスタ 32-5) を使って、アドレス比較動作中に SSPSR レジスタに格納された値をマスクできます。SSPxMSK レジスタのゼロ ('0') ビットがマスクビットとなり、受信アドレスの対応ビットは「ドントケア」として処理されます。

このレジスタは、全てのリセット条件で全ビットが「1」にリセットされるため、ユーザがマスク値を書き込まない限り、通常の SSP 動作には影響を与えません。

SSP マスクレジスタは、以下の時にアクティブ状態です。

- 7 ビットアドレス モード: A<7:1> のアドレス比較時
- 10 ビットアドレス モード: A<7:0> のアドレス比較時のみ。SSP マスクは、1 番目の (上位) バイトアドレスの受信中は無効

32.6 I²C マスタモード

マスタモードを有効にするには、SSPxCON1 レジスタの対応する SSPM ビットをセット/クリアして、SSPEN ビットをセットします。マスタモードでは SDA ピンと SCK ピンを入力として設定する必要があります。ピンを Low に駆動する必要がある場合、MSSP 周辺ハードウェアは出力ドライバの TRIS 制御より優先されます。

マスタモードの動作は、スタート/ストップ条件の検出による割り込み生成でサポートされます。ストップ (P) ビットとスタート (S) ビットは、リセット時または MSSP モジュールが無効にされた時にクリアされます。I²C バスの制御は、P ビットがセットされている場合またはバスがアイドル状態の場合に取得できます。

ファームウェア制御のマスタモードでは、ユーザコードがスタート/ストップビット条件の検出に基づいて、全ての I²C バス動作を実行します。スタート/ストップ条件の検出は、このモードで唯一能動的に動作する回路です。その他、全ての通信動作は、ユーザ ソフトウェアで SDA/SCL ラインを直接制御して実行されます。

以下のイベントが生じると、SSP 割り込みフラグビット (SSPxIF) がセットされます (SSP 割り込み有効時)。

- スタート条件の検出
- ストップ条件の検出
- データ転送バイトの送受信
- ACK の送受信
- 反復スタート条件の生成

Note 1: I²C マスタモードの場合、MSSP モジュールではイベントのキュー(待ち行列)処理ができません。例えばスタート条件を開始した直後(スタート条件が完了する前)に SSPxBUF レジスタへの書き込みを実行して送信動作を開始する事はできません。この場合 SSPxBUF への書き込みは実行されず、書き込みコリジョン検出ビット (WCOL) がセットされます。

2: マスタモードの場合、SEN/PEN ビットがクリアされるとスタート/ストップ検出がマスクされて割り込みが生成されます。これで割り込み生成が完了です。

32.6.1 I²C マスタモードの動作

マスタデバイスは、全てのシリアルクロックパルスとスタート/ストップ条件を生成します。転送は、ストップ条件または反復スタート条件で終了します。反復スタート条件は次のシリアル転送のスタート条件でもあるため、I²C バスは解放されません。

マスタ送信モードでは、SDA からシリアルデータを出し、SCL からシリアルクロックを出力します。最初の送信バイトは、受信デバイスのスレーブアドレス (7 ビット) と、読み書き (R/W) ビットを格納します。この場合、R/W ビットは論理「0」です。シリアルデータは、8 ビットずつ送信します。1 バイト送信ごとに ACK ビットを受信します。シリアル転送の開始/終了は、スタート/ストップ条件の出力で示します。

マスタ受信モードの場合、最初に送信するバイトは送信デバイスのスレーブアドレス (7 ビット) と R/W ビットを含みます。この場合、R/W ビットは論理「1」です。従って、最初の送信バイトは 7 ビットのスレーブアドレスの後に受信を示す「1」を格納します。シリアルデータは SDA で受信し、SCL はシリアルクロックを出力します。シリアルデータは、8 ビットずつ受信します。1 バイト受信ごとに ACK ビットを送信します。スタート条件とストップ条件は、転送の開始と終了を示します。

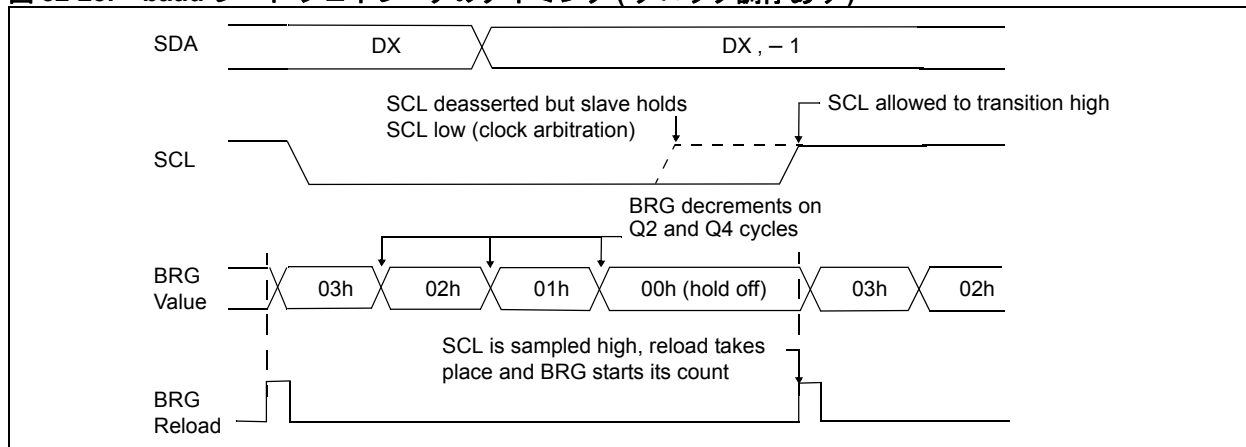
baud レート ジェネレータを使って、SCL のクロック周波数出力を設定します。詳細は [セクション 32.7 「baud レート ジェネレータ」](#) を参照してください。

PIC16(L)F1764/5/8/9

32.6.2 クロック調停

送受信、反復スタート/ストップ条件中にマスタが SCL ピンを解放 (SCL が High にプルアップ) すると、クロック調停が実行されます。SCL ピンが解放されても、実際に High がサンプリングされるまで baud レートジェネレータ (BRG) のカウントは中断します。SCL ピンで High がサンプリングされると、baud レートジェネレータに SSPxADD<7:0> の内容が再書き込みされてカウントを開始します。これにより、外部デバイスによってクロックが Low に保持されていたとしても、SCL の High 期間が最短でも BRG のロールオーバー1回分確保されます (図 32-25)。

図 32-25: baud レートジェネレータのタイミング (クロック調停あり)



32.6.3 WCOL ステータスフラグ

スタート、反復スタート、ストップ、受信、送信のいずれかのシーケンスが進行中にユーザが SSPxBUF に書き込もうとしても、WCOL がセットされてバッファの内容は変更されません (書き込みは実行されない)。このように、モジュールがアイドル状態でない時に SSPxBUF に対する何らかの操作をしようとすると WCOL ビットがセットされます。

Note: イベントのキューイングはできないため、スタート条件が完了するまで SSPxCON2 の下位 5 ビットへの書き込みは無効です。

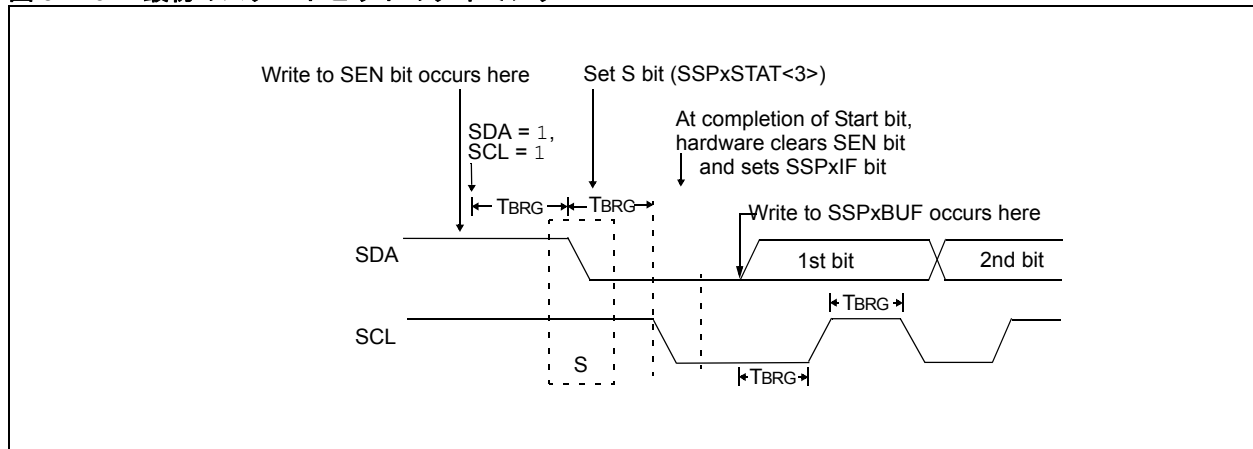
32.6.4 I²C マスタモードでのスタート条件のタイミング

スタート条件 (図 32-26) を開始するには、ユーザがスタートイネーブルビット (SSPxCON2レジスタのSENビット) をセットします。SDAピンとSCLピンがHighとしてサンプリングされると、baudレートジェネレータに SSPxADD<7:0> の内容が再書き込みされてカウントを開始します。baudレートジェネレータがタイムアウト (TBRG) した時にSCLとSDAの両方でHighがサンプルされると、SDAピンがLowに駆動されます。SCLピンがHighの時にSDAがLowに駆動されるとスタート条件となり、SSPxSTAT1レジスタのSビットがセットされます。続いて、baudレートジェネレータに SSPxADDM<7:0> の内容が再書き込みされ、カウントが再開します。baudレートジェネレータがタイムアウト (TBRG) すると、SSPxCON2レジスタのSENビットがハードウェアで自動的にクリアされ、baudレートジェネレータが一時停止状態になります。これによってSDAラインがLowに保持されスタート条件が完了します。

Note 1: スタート条件の開始時にSDAピンとSCLピンが既にLowにサンプルされている場合、またはスタート条件中にSDAラインがLowに駆動される前にSCLラインがLowとしてサンプルされた場合、バスコリジョンが生じてバスコリジョン割り込みフラグ (BCLIF) がセットされ、スタート条件が中止されてI²Cモジュールはアイドル状態にリセットされます。

2: Philips社のI²C™仕様書では、スタート時にバスコリジョンは発生しないと記載されています。

図 32-26: 最初のスタートビットのタイミング



PIC16(L)F1764/5/8/9

32.6.5 I²C マスタモードでの反復スタート条件のタイミング

SSPxCON2 レジスタの RSEN ビットが High で、マスタ ステートマシンがアクティブでないと、反復スタート条件 (図 32-27) が発生します。RSEN ビットがセットされると、SCL ピンが Low にアサートされます。SCL ピンで Low がサンプリングされると、baud レート ジェネレータに値が書き込まれてカウントが始まります。baud レート ジェネレータの 1 ロールオーバー カウント (TBRG) の間、SDA ピンは解放されます (High に遷移)。baud レート ジェネレータ タイムアウト時、SDA で High がサンプリングされると、SCL ピンがネゲートされます (High に遷移)。SCL ピンが High としてサンプリングされると、baud レート ジェネレータに値が再書き込みされてカウントを開始します。SDA と SCL は、1 TBRG の間、High としてサンプルされる必要があります。この動作の次には、SCL が High の間に、1 TBRG の期間、SDA ピンがアサートされます (SDA = 0)。SCL は Low にアサートされます。続いて、

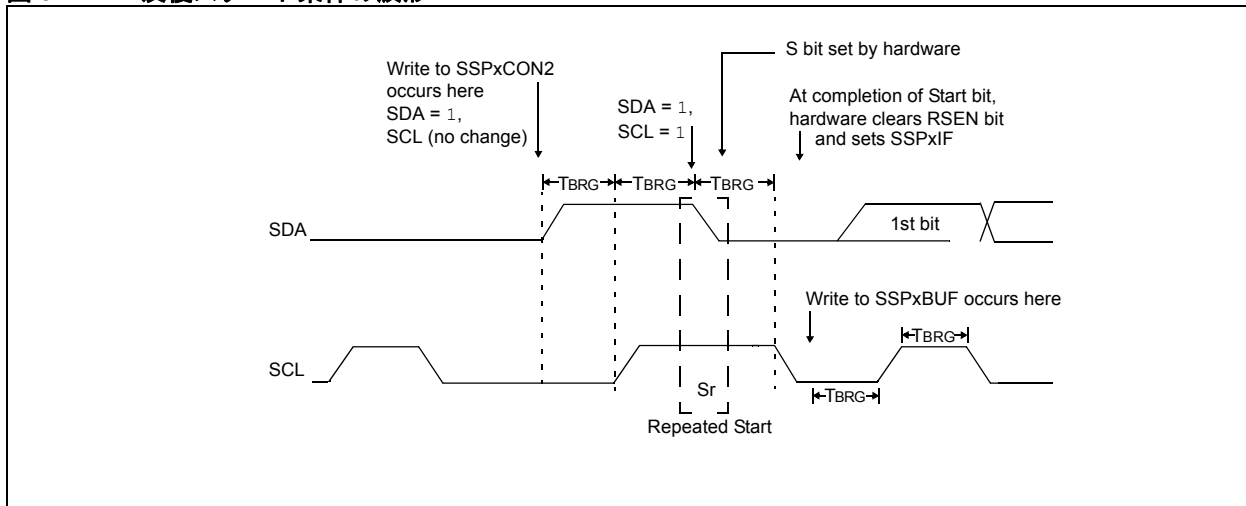
SSPxCON2 レジスタの RSEN ビットが自動的にクリアされる事で、baud レート ジェネレータには値が再度書き込まれず、SDA ピンが Low に保持されます。SDA ピンと SCL ピンでスタート条件が検出されるとすぐに、SSPxSTAT レジスタの S ビットがセットされます。SSPxIF ビットは、baud レート ジェネレータがタイムアウトするまでセットされません。

Note 1: その他のいかなるイベントが進行している間に RSEN がセットされても、実行されません。

2: 以下の場合、反復スタート条件中にバスコリジョンが発生します。

- SCL が Low から High に遷移した時、SDA が Low としてサンプリングされた。
- SDA が Low にアサートされる前に SCL が Low に遷移した (この場合、別のマスタがデータ「1」を送信しようとしている可能性があります)。

図 32-27: 反復スタート条件の波形



32.6.6 I²C マスタモード送信

データバイト、7 ビットアドレス、10 ビットアドレスの第 2 バイトは、SSPxBUF レジスタに値を書き込むだけで送信されます。書き込み動作によってバッファフルフラグビット BF がセットされ、baud レートジェネレータのカウントと共に次の送信が開始します。アドレス / データの各ビットは、SCL の立ち下がリエッジがアサートされた後に SDA ピン上にシフト出力されます。SCL は、1 回の BRG ロールオーバー (TBRG) の間 Low に保持されます。データは、SCL が High に解放される前に有効になる必要があります。SCL ピンは High に解放された後、TBRG の間、その状態を保持します。SDA ピン上のデータは、この期間および SCL の次の立ち下がリエッジ後のホールド時間まで安定している必要があります。8 番目のビットのシフト出力後 (8 番目のクロックの立ち下がリエッジ後)、BF フラグがクリアされマスタは SDA を解放します。アドレスが一致した場合、またはデータを正しく受信した場合、スレーブデバイスは 9 番目のビット時間中に ACK ビットを返す事ができます。9 番目のクロックの立ち上がりエッジで ACK のステータスが ACKSTAT ビットに書き込まれます。マスタが ACK を受信すると、ACKステータスビット (ACKSTAT) はクリアされます。受信しない場合、セットされたままです。9 番目のクロックの後、SSPxIF ビットがセットされ、マスタクロック (baud レートジェネレータ) は次のデータバイトが SSPxBUF に書き込まれるまで一時停止し、SCL は Low、SDA はそのままの状態を保持します (図 32-28)。

SSPxBUF への書き込み後、7 ビットのアドレスと R/W ビットが全て出力されるまで、SCL の立ち下がリエッジでアドレスのビットが 1 つずつシフト出力されます。8 番目のクロックの立ち下がリエッジで、マスタが SDA ピンを解放するとスレーブが肯定応答 (ACK) します。9 番目のクロックの立ち下がリエッジでマスタは SDA ピンをサンプリングし、スレーブがアドレスを認識できたかどうかを確認します。ACK ビットのステータスは、SSPxCON2 レジスタの ACKSTAT ステータスビットに書き込まれます。アドレス送信の 9 番目のクロック立ち下がリエッジ後、SSPxIF がセットされ BF フラグがクリアされ、次に SSPxBUF への書き込みが実行されるまで baud レートジェネレータは OFF となります (SCL は Low に保持され、SDA はフローティング)。

32.6.6.1 BF ステータスフラグ

送信モードの場合、SSPxSTAT レジスタの BF ビットは、CPU が SSPxBUF に書き込むとセットされ、8 ビットが全てシフト出力された時点でクリアされます。

32.6.6.2 WCOL ステータスフラグ

送信中 (SSPSR がデータバイトをシフト出力している途中) に、ユーザが SSPxBUF へ書き込もうとしても、WCOL がセットされバッファの内容は変更されません (書き込みは実行されない)。

次の送信の前に WCOL をソフトウェアでクリアする必要があります。

32.6.6.3 ACKSTAT ステータスフラグ

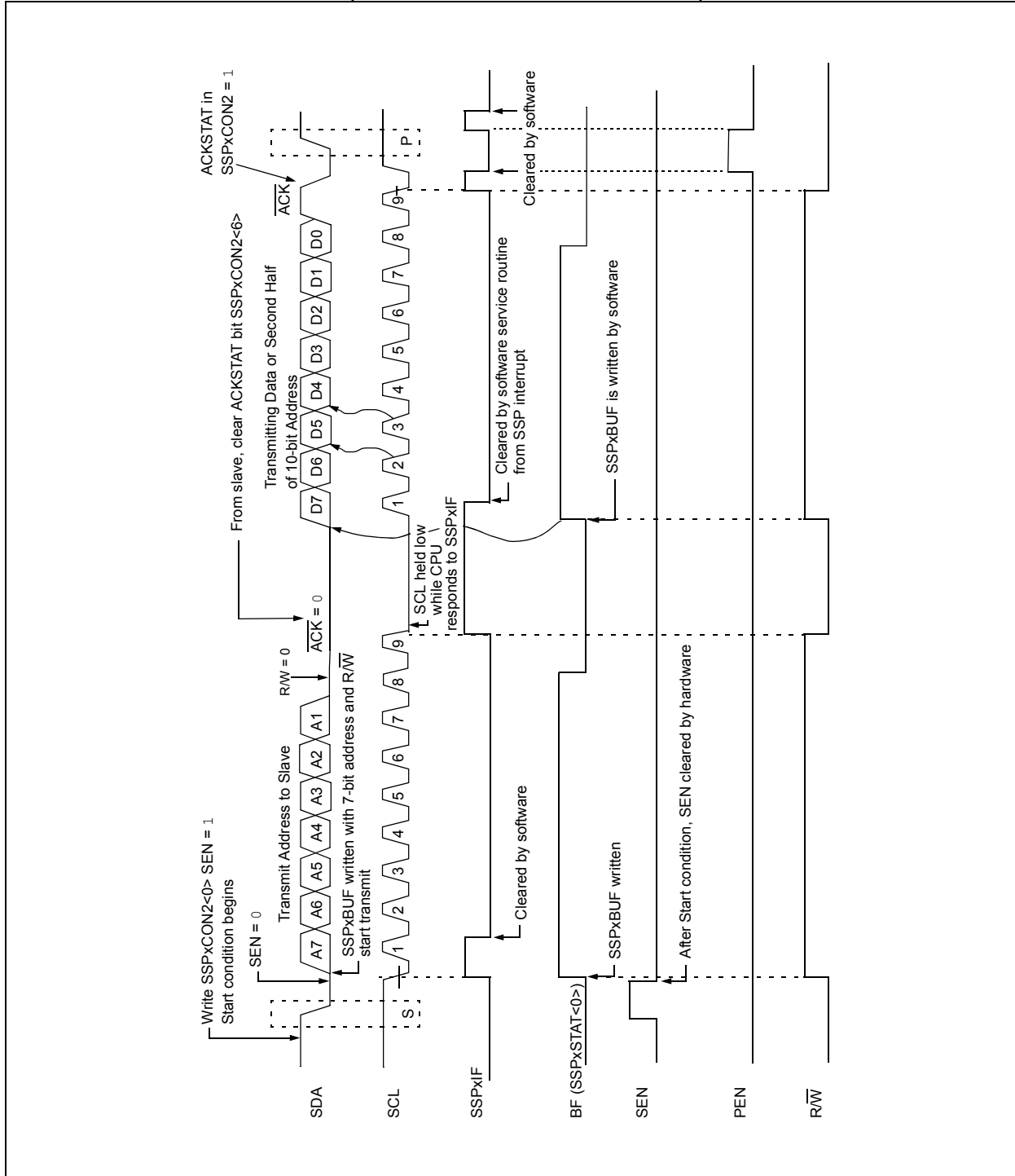
送信モードでは、スレーブが $\overline{\text{ACK}} = 0$ を送信すると SSPxCON2 レジスタの ACKSTAT ビットはクリアされ、スレーブが肯定応答しない ($\text{ACK} = 1$) とセットされます。スレーブは、自分自身のアドレスを認識するか (ジェネラルコールを含む)、正常にデータ受信を完了すると ACK を送信します。

32.6.6.4 代表的な送信シーケンス

1. ユーザが、SSPxCON2 レジスタの SEN ビットをセットしてスタート条件を生成する。
2. スタート条件の完了時に SSPxIF ビットがハードウェアによりセットされる。
3. ソフトウェアで SSPxIF ビットがクリアされる。
4. MSSP モジュールは、新しい動作を開始する前に必要なスタート時間が経過するまで待機する。
5. ユーザは送信先のスレーブアドレスを SSPxBUF に書き込む。
6. 8 ビット全てが送信されるまで、アドレスが SDA ピンにシフト出力される。SSPxBUF への書き込みが完了すると、送信が開始する。
7. MSSP モジュールは、スレーブデバイスから $\overline{\text{ACK}}$ ビットを受信して、その値を SSPxCON2 レジスタの ACKSTAT ビットに書き込む。
8. MSSP モジュールは、9 番目のクロックサイクルの最後に SSPxIF ビットをセットして割り込みを生成する。
9. ユーザは SSPxBUF に 8 ビットのデータを書き込む。
10. 8 ビット全てが送信されるまで、データが SDA ピンにシフト出力される。
11. MSSP モジュールは、スレーブデバイスから $\overline{\text{ACK}}$ ビットを受信して、その値を SSPxCON2 レジスタの ACKSTAT ビットに書き込む。
12. 全ての送信データバイトに対して、ステップ 8 ~ 11 を繰り返す。
13. ユーザが、SSPxCON2 レジスタの PEN または RSEN ビットをセットしてストップまたは反復スタート条件を生成する。ストップ / 反復スタート条件が完了すると、割り込みが生成される。

PIC16(L)F1764/5/8/9

図 32-28: I²C マスタモードの波形 (送信、7 ビット/10 ビットアドレス)



32.6.7 I²C マスタモード受信

マスタモード受信 (図 32-29) を有効にするには、受信イネーブルビット (SSPxCON2 レジスタの RCEN ビット) をプログラムします。

Note: RCEN ビットをセットする前に MSSP モジュールをアイドル状態にする必要があります。アイドル状態にない場合、RCEN ビットは無視されます。

baud レート ジェネレータがカウントを開始し、ロールオーバーごとに SCL ピンが遷移 (High → Low/Low → High) してデータが SSPSR へシフト入力されます。8 番目のクロックの立ち下がりエッジの後、受信イネーブルフラグが自動的にクリアされます。これによって SSPSR の内容が SSPxBUF に書き込まれ、BF フラグビットと SSPxIF フラグビットがセットされる事で、baud レート ジェネレータがカウントを停止して SCL が Low に保持されます。これによって MSSP はアイドル状態に移行し、次のコマンドを待ちます。CPU がバッファを読み出すと、BF フラグビットは自動的にクリアされます。ユーザは、受信の最後に肯定応答 (ACK) シーケンス イネーブル (SSPxCON2 レジスタの ACKEN ビット) をセットする事によって、肯定応答ビットを送信できます。

32.6.7.1 BF ステータスフラグ

受信動作の場合、SSPSR から SSPxBUF へアドレスバイトまたはデータバイトが書き込まれると、BF ビットがセットされます。このビットは SSPxBUF レジスタを読み出すとクリアされます。

32.6.7.2 SSPOV ステータスフラグ

受信動作の場合、以前の受信で BF フラグが既にセットされている時に SSPSR に 8 ビットが受信されると、SSPOV ビットがセットされます。

32.6.7.3 WCOL ステータスフラグ

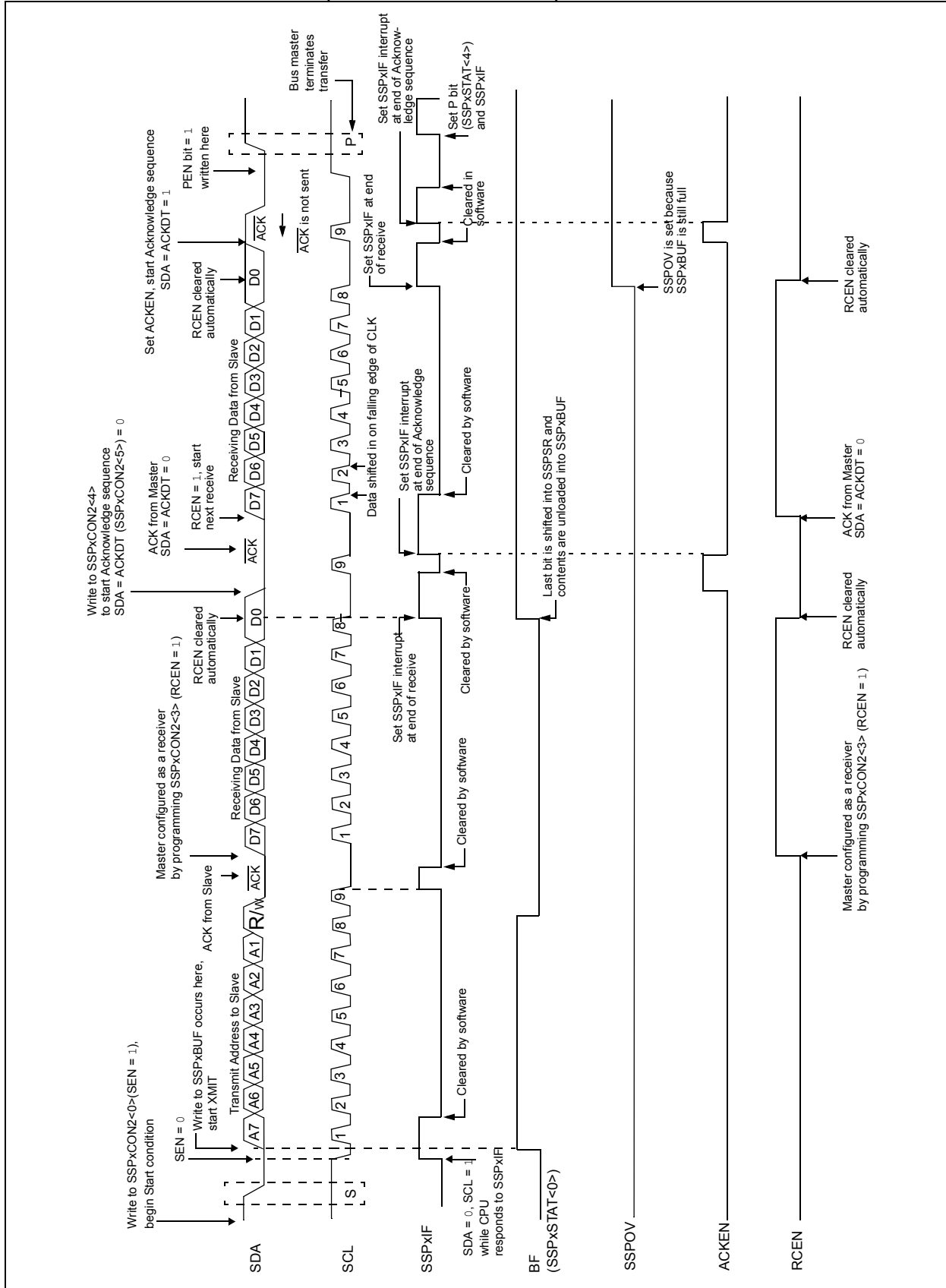
受信中 (SSPSR にデータバイトをシフト入力している途中) に、ユーザが SSPxBUF へ書き込もうとしても、WCOL がセットされ、バッファの内容は変更されません (書き込みは実行されない)。

32.6.7.4 代表的な受信シーケンス

1. ユーザが、SSPxCON2 レジスタの SEN ビットをセットしてスタート条件を生成する。
2. スタート条件の完了時に SSPxIF ビットがハードウェアによりセットされる。
3. ソフトウェアで SSPxIF ビットがクリアされる。
4. ユーザが送信先スレーブアドレスを SSPxBUF に書き込み、R/W ビットがセットされる。
5. 8 ビット全てが送信されるまで、アドレスが SDA ピンにシフト出力される。SSPxBUF への書き込みが完了すると、送信が開始する。
6. MSSP モジュールは、スレーブデバイスから $\overline{\text{ACK}}$ ビットを受信して、その値を SSPxCON2 レジスタの ACKSTAT ビットに書き込む。
7. MSSP モジュールは、9 番目のクロックサイクルの最後に SSPxIF ビットをセットして割り込みを生成する。
8. ユーザが SSPxCON2 レジスタの RCEN ビットをセットし、マスタがスレーブからバイトデータを受信する。
9. SCL の 8 番目の立ち下がりエッジ後、SSPxIF と BF がセットされる。
10. マスタが SSPxIF ビットをクリアし、SSPxBUF から受信バイトを読み出して BF ビットをクリアする。
11. マスタが、SSPxCON2 レジスタの ACKDT ビットを使って、スレーブへ送信する $\overline{\text{ACK}}$ 値を設定し、ACKEN ビットをセットして $\overline{\text{ACK}}$ を送信する。
12. マスタからスレーブへ $\overline{\text{ACK}}$ 信号が送信され、SSPxIF ビットがセットされる。
13. ユーザが SSPxIF をクリアする。
14. スレーブから 1 バイトを受信するたびに、ステップ 8 ~ 13 を繰り返す。
15. マスタが NOT $\overline{\text{ACK}}$ を送信するか、ストップ条件を送信すると通信が終了する。

PIC16(L)F1764/5/8/9

図 32-29: I²C マスタモードの波形 (受信、7 ビットアドレス)



32.6.8 肯定応答 (ACK) シーケンスのタイミング

肯定応答 (ACK) シーケンスを有効にするには、肯定応答シーケンスイネーブルビット (SSPxCON2 レジスタの ACKEN ビット) をセットします。このビットがセットされると SCL ピンが Low に駆動され、ACK データ (ACKDT) ビットの内容が SDA ピンに出力されます。ACK を生成するには、ACKDT ビットをクリアします。ACK を生成しない場合、ACK シーケンスが始まる前に ACKDT ビットをセットする必要があります。次に、baud レート ジェネレータが 1 ロールオーバー期間 (TBRG) カウントし、SCL ピンがネゲートされます (High に遷移)。SCL ピンが High がサンプルされた場合 (クロック調停)、baud レート ジェネレータは TBRG の期間カウントします。その後 SCL ピンが Low に駆動されます。続いて ACKEN ビットが自動的にクリアされ、baud レート ジェネレータが OFF になり、MSSP モジュールはアイドル状態に移行します (図 32-30)。

32.6.8.1 WCOL ステータスフラグ

ACK シーケンスの進行中にユーザが SSPxBUF に書き込もうとしても、WCOL がセットされ、バッファの内容は変更されません (書き込みは実行されない)。

32.6.9 ストップ条件のタイミング

ストップ シーケンス イネーブルビット (SSPxCON2 レジスタの PEN ビット) をセットすると、送受信終了時に SDA ピン上にストップビットがアサートされます。送受信終了時、9 番目のクロックの立ち下がりエッジの後に SCL ラインが Low に保持されます。PEN ビットがセットされると、マスタは SDA ラインを Low にアサートします。SDA ラインで Low がサンプリングされると、baud レート ジェネレータに値が再書き込みされ「0」までカウントダウンします。baud レート ジェネレータがタイムアウトすると、SCL ピンが High になり、TBRG の 1 周期 (baud レート ジェネレータのロールオーバー カウント) の間、SDA ピンがデアサートされます。SCL が High の間に SDA ピンが High としてサンプリングされると、SSPxSTAT レジスタの P ビットがセットされます。1 TBRG 経過後、PEN ビットはクリアされ、SSPxIF ビットがセットされます (図 32-31)。

32.6.9.1 WCOL ステータスフラグ

ストップ シーケンス実行中にユーザが SSPxBUF に書き込むと、WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されない)。

図 32-30: ACK シーケンスの波形

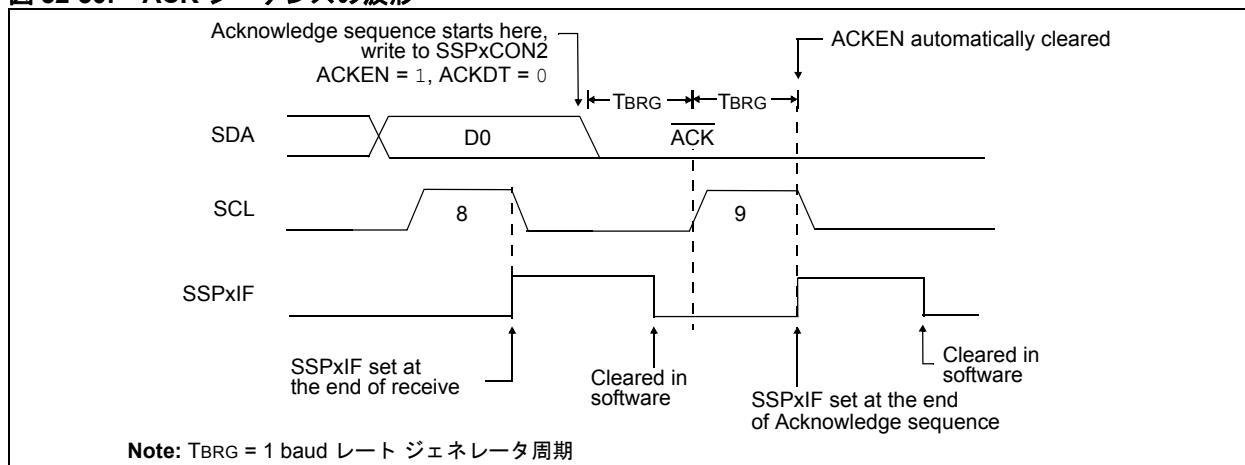
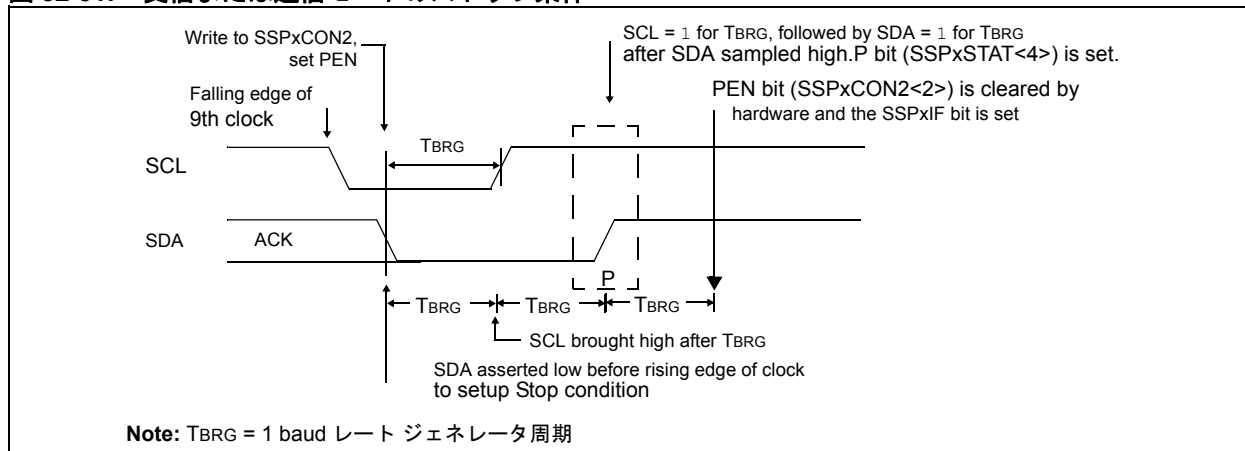


図 32-31: 受信または送信モードのストップ条件



PIC16(L)F1764/5/8/9

32.6.10 スリープ動作

スリープ中も、I²C スレーブ モジュールはアドレスまたはデータを受信可能であり、アドレスが一致した場合またはバイト転送が完了した場合、プロセッサをスリープから復帰できます (MSSP 割り込み有効時)。

32.6.11 リセットの影響

リセットは MSSP モジュールを無効にし、現在の転送は中止されます。

32.6.12 マルチマスタ モード

マルチマスタ モードでは、スタート/ストップ条件の検出によって生成される割り込みを基に、バスが空いている期間を判断します。ストップ (P) ビットとスタート (S) ビットは、リセット時または MSSP モジュールが無効にされた時にクリアされます。I²C バスの制御は、SSPxSTAT レジスタの P ビットがセットされた時、または S ビットと P ビットの両方がクリアされておりバスがアイドル状態の時に取得できます。バスがビジーの場合、SSP 割り込みを有効にすると、ストップ条件が発生した時に割り込みが生成されます。

マルチマスタ動作では、バス調停動作を実行できるように SDA ラインを監視して、信号レベルが期待される出力レベルにあるかどうかを確認する必要があります。確認はハードウェアが実行し、その結果は BCLIF ビットに格納されます。

下記のステートではバス調停に敗れる可能性があります。

- アドレス転送
- データ転送
- スタート条件
- 反復スタート条件
- ACK 条件

32.6.13 マルチマスタ通信、バスコリジョン、バス調停

マルチマスタ モードのサポートは、バス調停によって実現します。マスタがアドレス/データビットを SDA ピンに出力する際、マスタが SDA を High にプルアップして SDA に「1」を出力している間に別のマスタが「0」をアサートすると、調停が実行されます。SCL ピンが High にプルアップしている間、データは安定している必要があります。SDA ピンに表れるデータが「1」と予測される時に、SDA ピンで「0」がサンプリングされると、バスコリジョンが発生した事になります。マスタはバスコリジョン割り込みフラグ BCLIF をセットし、I²C ポートをアイドル状態にリセットします (図 32-32)。

送信動作中にバスコリジョンが発生した場合、送信動作が停止され、BF フラグがクリアされ、SDA ラインと SCL ラインがネゲートされ、SSPxBUF への書き込みが可能になります。ユーザがバスコリジョン割り込みサービスルーチンを使う場合、I²C バスが空いていれば、スタート条件をアサートする事で通信を再開できます。

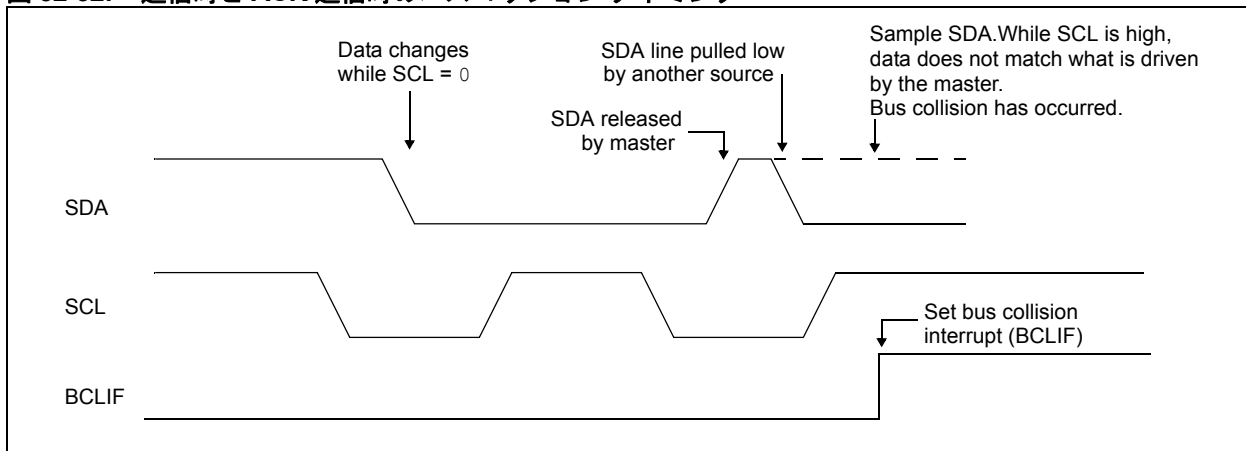
スタート、反復スタート、ストップ、ACK 条件中にバスコリジョンが発生した場合、これらの条件は中断され、SDA ラインと SCL ラインがネゲートされ、SSPxCON2 レジスタの対応する制御ビットがクリアされます。ユーザがバスコリジョン割り込みサービスルーチンを使う場合、I²C バスが空いていれば、スタート条件をアサートする事で通信を再開できます。

マスタは SDA ピンと SCL ピンの監視を続けます。ストップ条件が発生すると、SSPxIF ビットがセットされます。

SSPxBUF に書き込むと、バスコリジョン発生時にトランスミッタがどこまで送信を完了していたかに関わらず、データの先頭ビットから送信が始まります。

マルチマスタ モードでは、スタート/ストップ条件の検出によって生成される割り込みを基に、バスが空いている期間を判断します。I²C バスの制御は、SSPxSTAT レジスタの P ビットがセットされた場合、または S ビットと P ビットの両方がクリアされてバスがアイドル状態である場合に取得できます。

図 32-32: 送信時と ACK 送信時のバスコリジョン タイミング



32.6.13.1 スタート条件実行中のバスコリジョン

スタート条件中は、以下の場合にバスコリジョンが発生します。

- スタート条件の開始時に SDA または SCL で Low がサンプリングされた (図 32-33)。
- SDA が Low にアサートされる前に SCL で Low がサンプリングされた (図 32-34)。

スタート条件中、SDA ピンと SCL ピンは両方とも監視されています。

SDA ピンまたは SCL ピンが既に Low だった場合、以下の全てが実行されます。

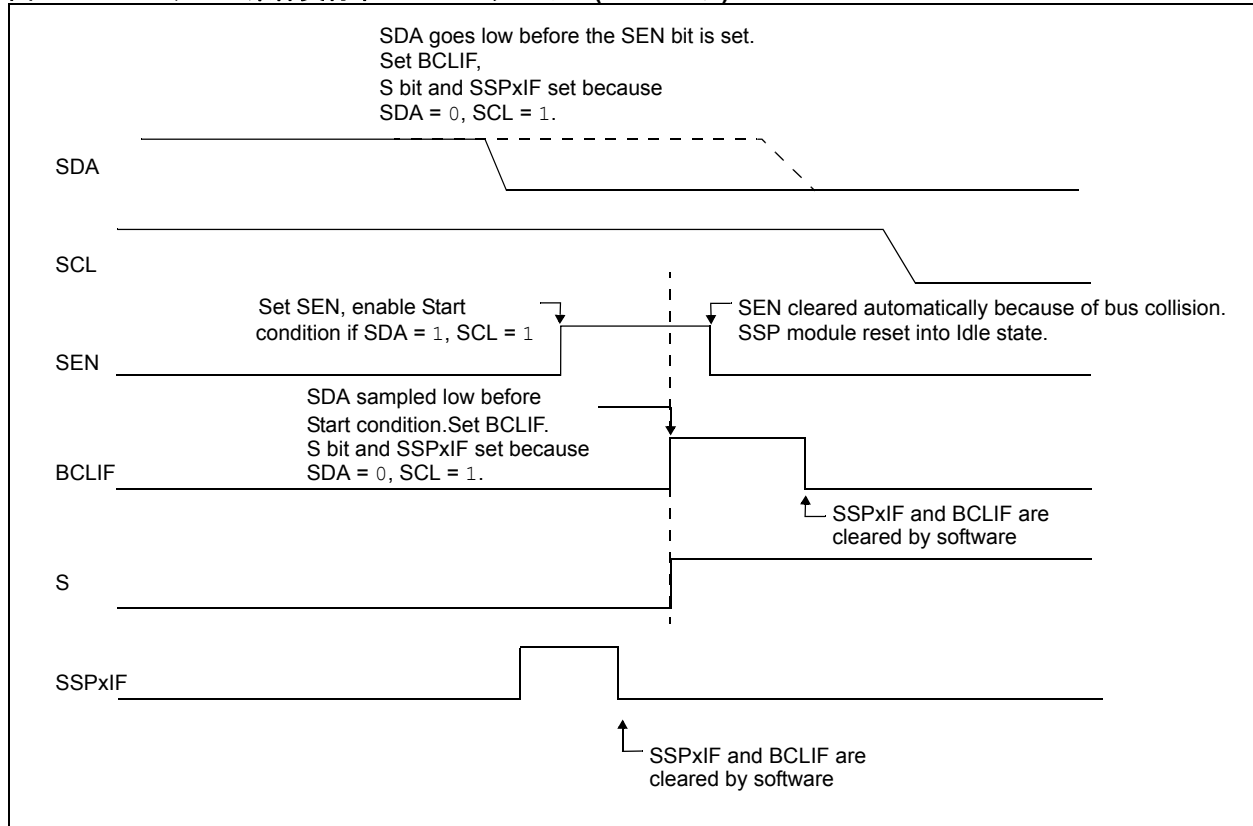
- スタート条件を中止する。
- BCLIF フラグをセットする。
- MSSP モジュールをアイドル状態にリセットする (図 32-33)。

スタート条件は、SDA ピンと SCL ピンのネゲートから始まります。SDA ピンで High がサンプリングされると、baud レート ジェネレータに値が書き込まれカウントダウンが始まります。SDA が High の時に SCL ピンが Low としてサンプルされた場合、スタート条件中に別のマスタがデータ「1」を駆動しようとしている事を意味するため、バスコリジョンが発生します。

上記の BRG カウント中に SDA ピンで Low がサンプリングされると、BRG はリセットされ、SDA ラインは BRG のタイムアウトを待たずにアサートされます (図 32-35)。しかし、SDA ピンで「1」がサンプリングされた場合、SDA ピンは BRG カウントの完了時に Low にアサートされます。次に baud レート ジェネレータが再書き込みされてゼロまでカウントダウンします。この間に SCL ピンで「0」がサンプリングされると、バスコリジョンは発生しません。BRG のカウント終了時、SCL ピンが Low にアサートされます。

Note: スタート条件中にバスコリジョンが問題にならないのは、2つのバスマスタが全く同じタイミングでスタート条件をアサートする事があり得ないためです。従って、必ずどちらか一方のマスタが他方よりも先に SDA をアサートします。この条件でバスコリジョンは発生しません。2つのマスタがスタート条件後の最初のアドレスで必ずバス調停できるためです。アドレスが同じ場合、さらにその後のデータ部分、反復スタート、ストップ条件で調停が必要です。

図 32-33: スタート条件実行中のバスコリジョン (SDA のみ)



PIC16(L)F1764/5/8/9

図 32-34: スタート条件実行中のバスコリジョン (SCL = 0)

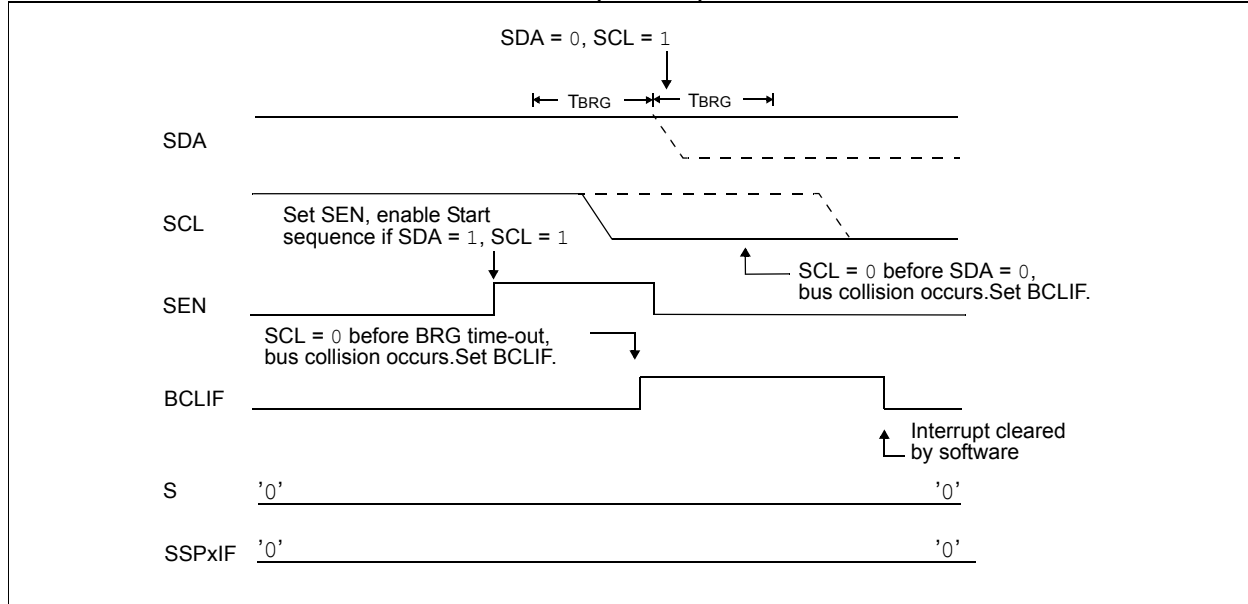
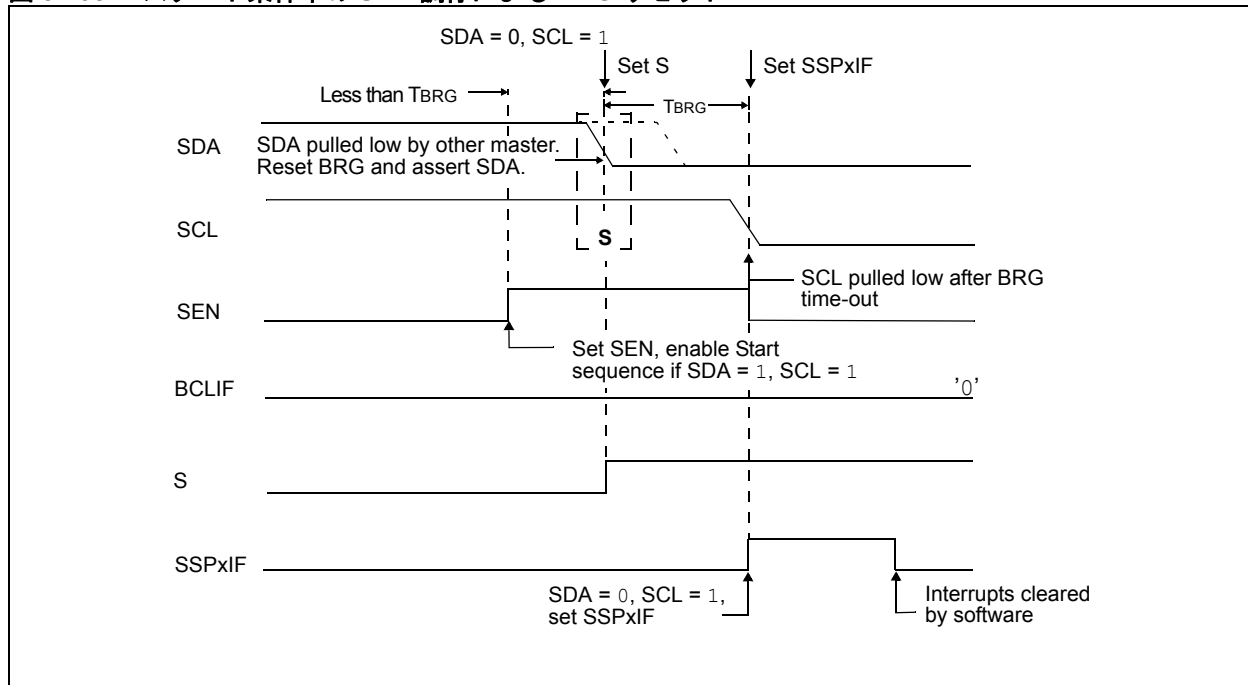


図 32-35: スタート条件中の SDA 調停による BRG リセット



32.6.13.2 反復スタート条件中のバスコリジョン

反復スタート条件中は、以下の場合にバスコリジョンが発生します。

- SCLがLowからHighに遷移する時にSDAでLowがサンプリングされる(ケース1)。
- SDAでLowがアサートされる前にSCLがLowに遷移した(他のマスタがデータ「1」を送信しようとしている事を示す)(ケース2)。

ユーザが SDA ピンを解放し、ピンが High にフローティングできるようになると、BRG に SSPxADD の内容が書き込まれてゼロまでカウントダウンします。続いて SCL ピンがネゲートされ、High がサンプリングされると、SDA ピンがサンプリングされます。

SDA が Low であれば、バスコリジョンが発生しています(つまり、別のマスタがデータ「0」を送信しようとしている、[図 32-36](#))。SDA で High がサンプリングされた場合、BRG が再度書き込まれカウントを開始します。BRG がタイムアウトする前に SDA が High から Low に遷移した場合、バスコリジョンは発生しません。2つのマスタが全く同じタイミングで SDA をアサートする事はあり得ないからです。

BRG がタイムアウトする前に SCL が High から Low に遷移し、SDA がまだアサートされていない場合はバスコリジョンが発生します。この場合、反復スタート条件実行中に、他のマスタがデータ「1」を送信しようとしています([図 32-37](#) 参照)。

BRG タイムアウト時に SCL と SDA の両方が High のままの場合、SDA ピンが Low に駆動され、BRG の再書き込み後にカウントを開始します。カウントが終了すると、SCL ピンはその状態に関係なく Low に駆動され、反復スタート条件が終了します。

図 32-36: 反復スタート条件中のバスコリジョン(ケース1)

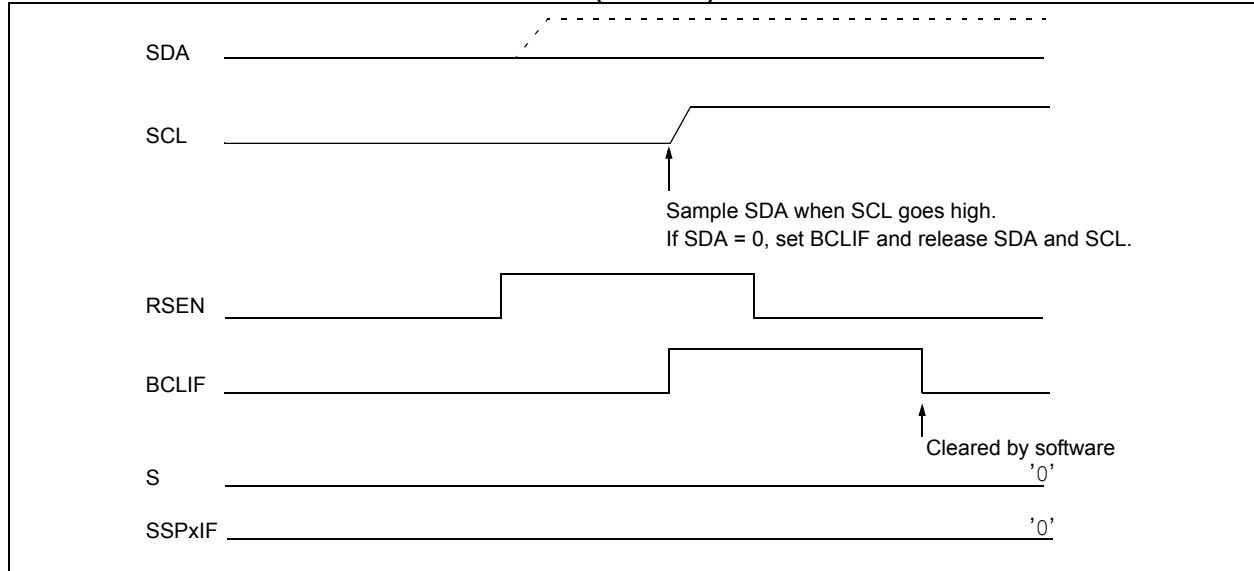
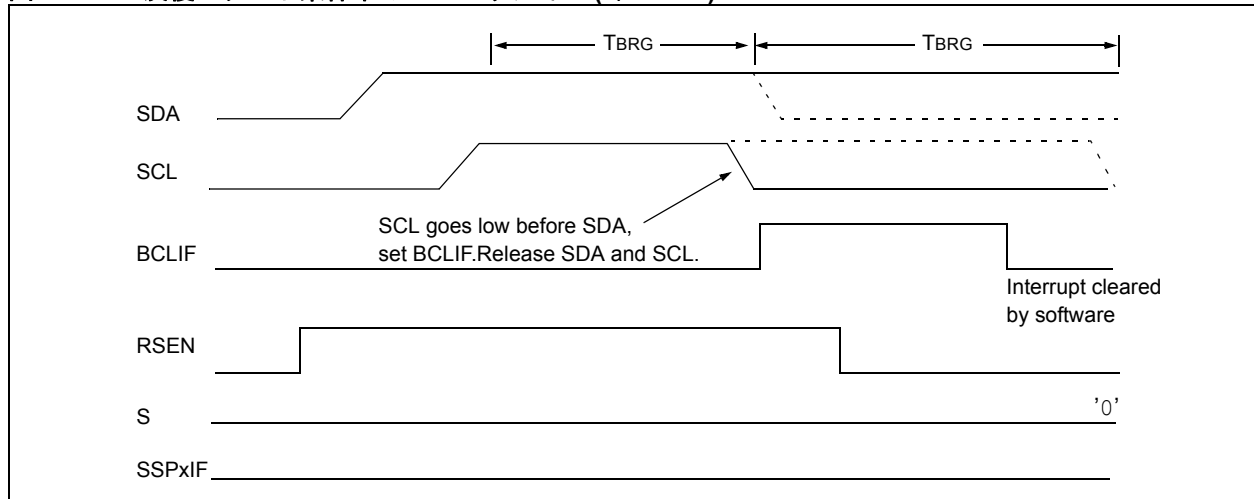


図 32-37: 反復スタート条件中のバスコリジョン(ケース2)



PIC16(L)F1764/5/8/9

32.6.13.3 ストップ条件中のバスコリジョン

ストップ条件中は、以下の場合にバスコリジョンが発生します。

- a) SDA ピンがネゲートされてフローティング High が可能になった後、BRGのタイムアウト後に SDA で Low がサンプリングされる (ケース 1)。
- b) SCL ピンがネゲートされた後、SDA が High に遷移する前に SCL で Low がサンプリングされる (ケース 2)。

ストップ条件は、SDA の Low アサートから開始します。SDA で Low がサンプルされると、SCL ピンをフローティング状態にできます。SCL ピンで High がサンプ

リングされると (クロック調停)、baud レート ジェネレータに SSPxADD の値が書き込まれ 0 へのカウントダウンが始まります。BRG のタイムアウト後、SDA がサンプリングされます。SDA で Low がサンプリングされた場合、バスコリジョンが発生しています。これは、別のマスタがデータ「0」を駆動しようとしているためです (図 32-38)。SDA が High にフローティングできるようにする前に SCL ピンで Low がサンプリングされると、バスコリジョンが発生します。これは、別のマスタがデータ「0」を駆動しようとしているもう 1 つのケースです (図 32-39)。

図 32-38: ストップ条件中のバスコリジョン (ケース 1)

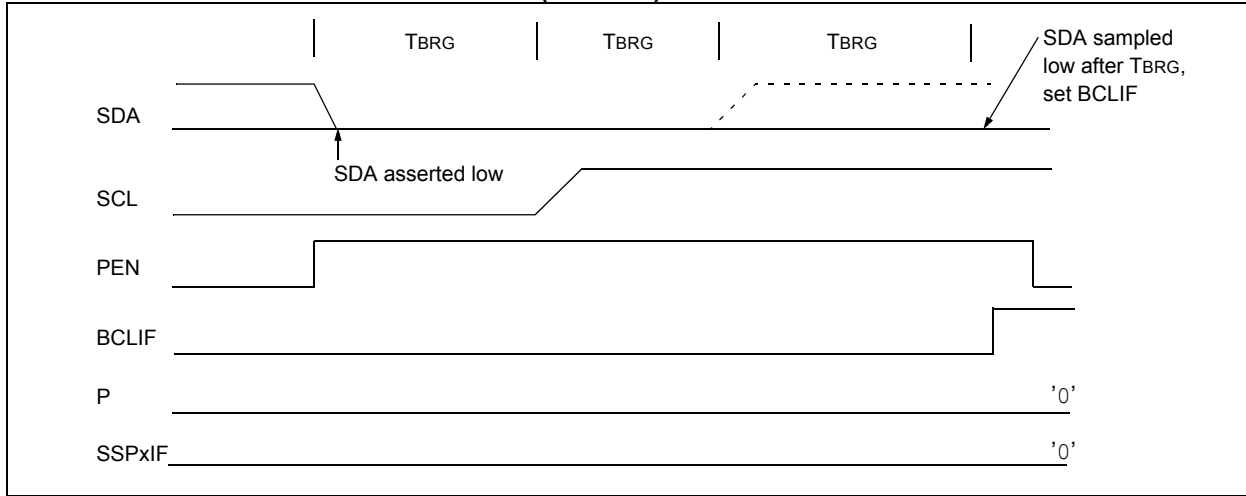


図 32-39: ストップ条件中のバスコリジョン (ケース 2)

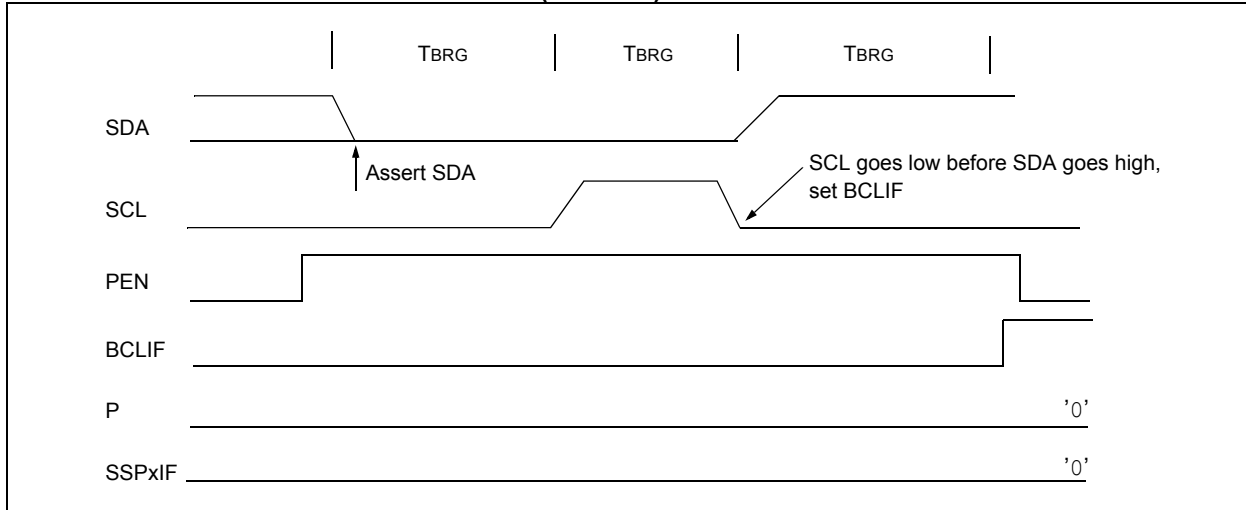


表 32-3: I²C™ 動作関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
RxyPPS	—	—	—	RxyPPS<4:0>					158
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					158、160
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					158、160
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					158、160
SSP1ADD	ADD<7:0>								427
SSP1BUF	同期シリアルポート受信 / 送信バッファレジスタ								379*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				424
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	425
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	426
SSP1MSK	MSK<7:0>								427
SSP1STAT	SMP	CKE	D \bar{A}	P	S	R \bar{W}	UA	BF	423
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

凡例: — = 未実装、「0」として読み出し。網掛けの部分は I²C™ モードの MSSP モジュールでは使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

PIC16(L)F1764/5/8/9

32.7 BAUD レート ジェネレータ

MSSP モジュールは baud レート ジェネレータを装備しており、I²C と SPI マスタの両方のモードでクロックの生成が可能です。baud レート ジェネレータ (BRG) の再書き込み値は、SSPxADD レジスタの値です (レジスタ 32-6 参照)。SSPxBUF への書き込みが発生すると、baud レート ジェネレータが自動的にカウントダウンを開始します。

動作が完了すると内部クロックは自動的にカウントを停止し、クロックピンはそのままの状態を保持します。

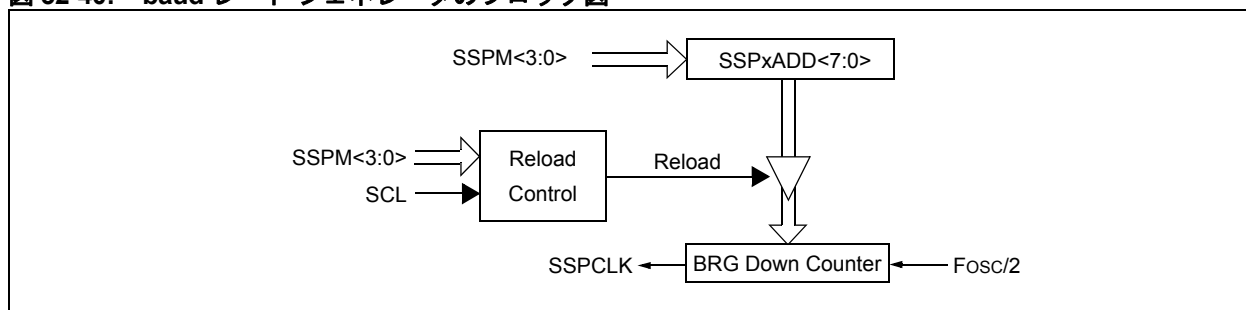
内部信号「Reload (図 32-40 参照)」によって、SSPxADD の値が BRG カウンタに書き込まれます。この動作はモジュールのクロックラインの各周期で 2 回発生します。再書き込み信号がアサートされるタイミングは、MSSP の動作モードで決まります。

表 32-4 に、命令サイクルに基づくクロックレートと SSPxADD に再書き込みされる BRG の値を示します。

式 32-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

図 32-40: baud レート ジェネレータのブロック図



Note: I²C モードの baud レート ジェネレータとして使う場合、0x00、0x01、0x02 は SSPxADD の値として無効です。これは、実装上の制限です。

表 32-4: MSSP クロックレートと BRG の値

Fosc	Fcy	BRG 値	Fclock (BRG ロールオーバー 2 回)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: システムが IOL 要件をサポートするように設計するには、表 36-4 の I/O ポートの電氣的仕様を参照してください。

32.8 レジスタ定義 : MSSP 制御

レジスタ 32-1: SSP1STAT: SSP ステータス レジスタ

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

凡例 :

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7	<p>SMP: SPI データ入力サンプリング ビット</p> <p><u>SPI マスタモード:</u> 1 = データ出力時間の最後に入力データをサンプリングする 0 = データ出力時間の中間で入力データをサンプリングする</p> <p><u>SPI スレーブモード:</u> SPI がスレーブモードの場合、SMP をクリアする必要があります。</p> <p><u>I²C マスタまたはスレーブモード:</u> 1 = スルーレート制御の標準速度モード (100 kHz と 1 MHz) を無効にする 0 = スルーレート制御の高速モード (400 kHz) を有効にする</p>
bit 6	<p>CKE: SPI クロックエッジ選択ビット (SPI モードのみ)</p> <p><u>SPI マスタまたはスレーブモード:</u> 1 = クロックの状態がアクティブからアイドルに遷移する時に送信する 0 = クロックの状態がアイドルからアクティブに遷移する時に送信する</p> <p><u>I²CTM モードのみ:</u> 1 = SMBus 仕様に準拠したしきい値の入力ロジックを有効にする 0 = SMBus 仕様入力を無効にする</p>
bit 5	<p>D/A: Data/Address ビット (I²C モードのみ)</p> <p>1 = 最後に送受信したバイトがデータであることを示す 0 = 最後に送受信したバイトがアドレスであることを示す</p>
bit 4	<p>P: ストップビット</p> <p>(I²C モードでのみ使います。SSPEN がクリアされ MSSP モジュールが無効になると、このビットはクリアされます。)</p> <p>1 = 最後にストップビットが検出された (このビットは、リセット時に「0」となる) 0 = 最後にストップビットは検出されていない</p>
bit 3	<p>S: スタートビット</p> <p>(I²C モードでのみ使います。SSPEN がクリアされ MSSP モジュールが無効になると、このビットはクリアされます。)</p> <p>1 = 最後にスタートビットが検出された (このビットは、リセット時に「0」となる) 0 = 最後にスタートビットは検出されていない</p>
bit 2	<p>R/W: 読み書きビット情報 (I²C モードのみ)</p> <p>このビットは、最後のアドレス一致後の R/W ビット情報を保持します。このビットは、アドレス一致が発生してからスタートビット、ストップビット、NOT ACK ビットのいずれかを受信するまでの期間でのみ有効です。</p> <p><u>I²C スレーブモード:</u> 1 = 読み出し 0 = 書き込み</p> <p><u>I²C マスタモード:</u> 1 = 送信中 0 = 送信中ではない</p> <p>このビットと SEN、RSEN、PEN、RCEN、ACKEN のいずれかを OR 演算すると、MSSP がアイドルであるかどうか分かります。</p>
bit 1	<p>UA: アドレス更新ビット (10 ビット I²C モードのみ)</p> <p>1 = ユーザが SSP1ADD レジスタ内のアドレスを更新する必要があることを示す 0 = アドレスの更新は不要</p>
bit 0	<p>BF: バッファフルステータスビット</p> <p><u>受信 (SPI および I²C モード):</u> 1 = 受信は完了した (SSP1BUF はフル) 0 = 受信は未完了で、SSP1BUF はエンプティである</p> <p><u>送信 (I²C モードのみ):</u> 1 = データ送信中 (ACK およびストップビットを含まない) で、SSP1BUF はフルである 0 = データ送信が完了 (ACK およびストップビットを含まない) し、SSP1BUF はエンプティである</p>

PIC16(L)F1764/5/8/9

レジスタ 32-2: SSP1CON1: SSP 制御レジスタ 1

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPOV ⁽¹⁾	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	HS = ビットはハードウェアでセット C = ビットはユーザがクリア

bit 7	<p>WCOL: 書き込みコリジョン検出ビット</p> <p>マスタモード:</p> <p>1 = I²C の送信開始条件が無効である時に SSP1BUF レジスタへの書き込みが試行された</p> <p>0 = 書き込みコリジョンは発生していない</p> <p>スレーブモード:</p> <p>1 = 前のワードの送信中に SSP1BUF レジスタへの書き込みが実行された (ソフトウェアでクリアする必要がある)</p> <p>0 = 書き込みコリジョンは発生していない</p>
bit 6	<p>SSPOV: 受信オーバーフローインジケータビット⁽¹⁾</p> <p>SPIモード:</p> <p>1 = SSP1BUF レジスタが前のバイトを保持中に次のバイトを受信した。オーバーフローが発生すると、SSPSR 内のデータは失われる。オーバーフローは、スレーブモードの場合のみ発生する。スレーブモードでオーバーフローの設定を回避するには、データ送信だけを実行する場合でも SSP1BUF を読み出す必要がある。マスタモードの場合、SSP1BUF レジスタへの書き込みによって新規の受信 (および送信) を開始するため、オーバーフロービットはセットされない (ソフトウェアでクリアする必要がある)</p> <p>0 = オーバーフローは発生していない</p> <p>I²Cモード:</p> <p>1 = SSP1BUF レジスタが前のバイトを保持中に次のバイトを受信した。送信モードの場合、SSPOV ビットは「ドントケア」 (ソフトウェアでクリアする必要がある)</p> <p>0 = オーバーフローは発生していない</p>
bit 5	<p>SSPEN: 同期シリアルポートイネーブルビット</p> <p>両モード共、これらのピンが有効の場合、入力または出力として適切に設定する必要があります。</p> <p>SPIモード:</p> <p>1 = シリアルポートを有効にして、SCK、SDO、SDI、\overline{SS} をシリアルポートピン⁽²⁾のソースとして設定する</p> <p>0 = シリアルポートを無効にし、上記のピンを I/O ポートピンとして設定する</p> <p>I²Cモード:</p> <p>1 = シリアルポートを有効にして、SDA ピンと SCL ピンをシリアルポートピン⁽³⁾のソースとして設定する</p> <p>0 = シリアルポートを無効にし、上記のピンを I/O ポートピンとして設定する</p>
bit 4	<p>CKP: クロック極性選択ビット</p> <p>SPIモード:</p> <p>1 = クロックのアイドル状態を High レベルに設定する</p> <p>0 = クロックのアイドル状態を Low レベルに設定する</p> <p>I²Cスレーブモード:</p> <p>SCL 解放制御</p> <p>1 = クロックを有効にする</p> <p>0 = クロックを Low に保持する (クロックストレッチ)。 (データのセットアップ時間確保のために使われる)</p> <p>I²Cマスタモード:</p> <p>このモードでは使わない</p>
bit 3-0	<p>SSPM<3:0>: 同期シリアルポートモード選択ビット</p> <p>1111 = I²C スレーブモード、10 ビットアドレス、スタートビットとストップビットの割り込みを有効にする</p> <p>1110 = I²C スレーブモード、7 ビットアドレス、スタートビットとストップビットの割り込みを有効にする</p> <p>1101 = 予約済み</p> <p>1100 = 予約済み</p> <p>1011 = I²C ファームウェア制御マスタモード (スレーブアイドル)</p> <p>1010 = SPI マスタモード、クロック = $F_{osc}/(4 * (SSP1ADD+1))^{(5)}$</p> <p>1001 = 予約済み</p> <p>1000 = I²C マスタモード、クロック = $F_{osc}/(4 * (SSP1ADD+1))^{(4)}$</p> <p>0111 = I²C スレーブモード、10 ビットアドレス</p> <p>0110 = I²C スレーブモード、7 ビットアドレス</p> <p>0101 = SPI スレーブモード、クロック = SCK ピン、\overline{SS} ピン制御は無効 (\overline{SS} ピンは I/O ピンとして使用可能)</p> <p>0100 = SPI スレーブモード、クロック = SCK ピン、\overline{SS} ピン制御有効化</p> <p>0011 = SPI マスタモード、クロック = $T2_match/2$</p> <p>0010 = SPI マスタモード、クロック = $F_{osc}/64$</p> <p>0001 = SPI マスタモード、クロック = $F_{osc}/16$</p> <p>0000 = SPI マスタモード、クロック = $F_{osc}/4$</p>

- Note**
- 1: マスタモードの場合、新たな送受信は SSP1BUF レジスタへの書き込みによって開始するため、オーバーフロービットはセットされません。
 - 2: 有効にする場合、これらのピンは入出力として適切に設定する必要があります。ピンを選択するには、SSPSSPPS、SSPCLKPPS、SSPDATPPS、RxyPPS を使います。
 - 3: 有効にする場合、SDA ピンと SCL ピンを入力として設定する必要があります。ピンを選択するには、SSPCLKPPS、SSPDATPPS、RxyPPS を使います。
 - 4: I²C モードの場合、SSP1ADD の値 0、1、2 はサポートされていません。
 - 5: SSP1ADD の値「0」はサポートされていません。代わりに SSPM = 0000 を使います。

レジスタ 32-3: SSP1CON2: SSP 制御レジスタ 2⁽¹⁾

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	HC = ハードウェアでクリア S = ビットはユーザがセット

bit 7	GCEN: ジェネラルコール イネーブルビット (I ² C スレーブモードのみ) 1 = SSPSR でジェネラルコール アドレス (0x00 または 00h) を受信した時の割り込みを有効にする 0 = ジェネラルコール アドレスを無効にする
bit 6	ACKSTAT: 肯定応答 (ACK) ステータスビット (I ² C モードのみ) 1 = ACK を受信していない 0 = ACK を受信した
bit 5	ACKDT: 肯定応答 (ACK) データビット (I ² C モードのみ) <u>受信モード:</u> 受信の最後にユーザが ACK シーケンスを開始する時に送信される値です。 1 = NOT ACK 0 = ACK
bit 4	ACKEN: 肯定応答 (ACK) シーケンス イネーブルビット (I ² C マスタモードのみ) <u>マスタ受信モード:</u> 1 = SDA ピンと SCL ピンで ACK シーケンスを開始し、ACKDT データビットを送信する。ハードウェアで自動的にクリアされる 0 = 肯定応答シーケンスを開始しない
bit 3	RCEN: 受信イネーブルビット (I ² C マスタモードのみ) 1 = I ² C の受信モードを有効にする 0 = 受信をアイドル状態にする
bit 2	PEN: ストップ条件イネーブルビット (I ² C マスタモードのみ) <u>SCKMSSP 解放制御:</u> 1 = SDA ピンと SCL ピンでストップ条件を開始する。ハードウェアで自動的にクリアされる 0 = ストップ条件を開始しない
bit 1	RSEN: 反復スタート条件イネーブルビット (I ² C マスタモードのみ) 1 = SDA ピンと SCL ピンで反復スタート条件を開始する。ハードウェアで自動的にクリアされる 0 = 反復スタート条件を開始しない
bit 0	SEN: スタート条件イネーブル / ストレッチ イネーブルビット <u>マスタモード:</u> 1 = SDA ピンと SCL ピンでスタート条件を開始する。ハードウェアで自動的にクリアされる 0 = スタート条件を開始しない <u>スレーブモード:</u> 1 = スレーブ送信とスレーブ受信の両方でクロック ストレッチを有効にする (ストレッチ イネーブル) 0 = クロック ストレッチを無効にする

Note 1: ACKEN、RCEN、PEN、RSEN、SEN ビットについて: I²C モジュールがアイドル状態でない場合、このビットがセットされず (スプール処理なし)、SSP1BUF に書き込まれない場合があります (SSP1BUF への書き込みは無効です)。

PIC16(L)F1764/5/8/9

レジスタ 32-4: SSP1CON3: SSP 制御レジスタ 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM ⁽³⁾	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

- bit 7 **ACKTIM:** 肯定応答 (ACK) タイムステータス ビット (I²C モードのみ)⁽³⁾
 1 = I²C バスは ACK シーケンスを実行中であることを示す。SCL クロックの 8 番目の立ち下がりエッジでセットされる
 0 = ACK シーケンスではない。SCL クロックの 9 番目の立ち上がりエッジでクリアされる
- bit 6 **PCIE:** ストップ条件割り込みイネーブルビット (I²C モードのみ)
 1 = ストップ条件検出時の割り込みを有効にする
 0 = ストップ条件検出時の割り込みを無効にする⁽²⁾
- bit 5 **SCIE:** スタート条件割り込みイネーブルビット (I²C モードのみ)
 1 = スタートまたは反復スタート条件検出時の割り込みを有効にする
 0 = ストップ条件検出時の割り込みを無効にする⁽²⁾
- bit 4 **BOEN:** バッファ上書きイネーブルビット
SPI スレーブモード:⁽¹⁾
 1 = SSP1BUF は、新しいデータバイトがシフト入力されるたびに更新される (BF ビットは無視する)
 0 = SSP1STAT レジスタの BF ビットが既にセットされている状態で新しいバイトを受信した場合、SSP1CON1 レジスタの SSPOV ビットがセットされ、バッファは更新されない
I²C マスタおよび SPI マスタモード:
 このビットを無視する
I²C スレーブモード:
 1 = SSP1BUF は更新され、受信したアドレス / データバイトに対して ACK 信号が生成される。BF ビットが 0 の場合のみ、SSPOV ビットの状態は無視される
 0 = SSPOV がクリアされている時のみ SSP1BUF が更新される
- bit 3 **SDAHT:** SDA ホールド時間の選択ビット (I²C モードのみ)
 1 = SDA のホールド時間を SCL の立ち下がりエッジから最小 300 ns にする
 0 = SDA のホールド時間を SCL の立ち下がりエッジから最小 100 ns にする
- bit 2 **SBCDE:** スレーブモードのバスコリジョン検出イネーブルビット (I²C スレーブモードのみ)
 SCL の立ち上がりエッジで、モジュール出力が High の場合に SDA で Low がサンプリングされると、PIR2 レジスタの BCL11F ビットがセットされてバスはアイドル状態になる。
 1 = スレーブ バスコリジョン割り込みを有効にする
 0 = スレーブ バスコリジョン割り込みを無効にする
- bit 1 **AHEN:** アドレスホールドイネーブルビット (I²C スレーブモード専用)
 1 = 一致受信アドレスバイトの SCL の 8 番目の立ち下がりエッジ後、SSP1CON1 レジスタの CKP ビットがクリアされ、SCL が Low に保持される
 0 = アドレスホールド機能を無効にする
- bit 0 **DHEN:** データホールドイネーブルビット (I²C スレーブモードのみ)
 1 = 受信データバイトについて SCL の 8 番目の立ち下がりエッジ後、スレーブ ハードウェアによって SSP1CON1 レジスタの CKP ビットがクリアされ、SCL が Low に保持される
 0 = データホールド機能を無効にする

- Note** 1: デイジーチェーン接続された SPI の動作です。ユーザは、最後の受信バイト以外は全て無視できます。新しいバイトが受信されても SSPOV はセットされたままであり、BF = 1 ですが、ハードウェアは最新のバイトを SSP1BUF へ書き込み続けます。
- 2: このビットは、スタート / ストップ条件検出を明示的に有効にしているスレーブモードには影響しません。
- 3: ACKTIM ステータスビットがアクティブになるのは、AHEN ビットまたは DHEN ビットがセットされている場合のみです。

レジスタ 32-5: SSP1MSK: SSP マスクレジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

bit 7-1 **MSK<7:1>**: マスクビット

- 1 = 受信アドレスの bit n を SSP1ADD<n> と比較して I²C アドレス一致を検出する
- 0 = 受信アドレスの bit n を I²C アドレス一致の検出に使わない

bit 0 **MSK<0>**: I²C スレーブモード、10 ビットアドレス時のマスクビット

- I²C スレーブモード、10 ビットアドレス (SSPM<3:0> = 0111 または 1111) の場合:
- 1 = 受信アドレスの bit 0 を SSP1ADD<0> と比較して I²C アドレス一致を検出する
- 0 = 受信アドレスの bit 0 を I²C アドレス一致の検出に使わない
- I²C スレーブモード、7 ビットアドレスでは、このビットは無視されます。

レジスタ 32-6: SSP1ADD: MSSP アドレス /baud レートレジスタ (I²C モード)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADD<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
u = ビットは不変	x = ビットは未知	-n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット	「0」= ビットはクリア	

マスタモード:

bit 7-0 **ADD<7:0>**: baud レートクロック分周ビット
 $SCL \text{ ピンクロック周期} = ((ADD<7:0> + 1) * 4) / F_{osc}$

10 ビットスレーブモード - 上位アドレスバイト:

bit 7-3 **未使用**: 上位アドレスバイトでは使わない。このレジスタのビット状態は「ドントケア」である。マスタが送信するビットパターンは I²C 仕様で定められており、「11110」とする必要がある。これらのビットに対してもハードウェアによる比較は実行されるが、このレジスタの値は比較結果に影響を与えない

bit 2-1 **ADD<2:1>**: 10 ビットアドレスの上位 2 ビット

bit 0 **未使用**: このモードでは使わない。ビット状態は「ドントケア」である

10 ビットスレーブモード - 下位アドレスバイト:

bit 7-0 **ADD<7:0>**: 10 ビットアドレスの下位 8 ビット

7 ビットスレーブモード:

bit 7-1 **ADD<7:1>**: 7 ビットのアドレス

bit 0 **未使用**: このモードでは使わない。ビット状態は「ドントケア」である

PIC16(L)F1764/5/8/9

33.0 拡張 USART (EUSART: Enhanced Universal Synchronous Asynchronous Receiver Transmitter)

拡張USART (EUSART: Enhanced Universal Synchronous Asynchronous Receiver Transmitter) モジュールは、シリアル I/O 通信用の周辺モジュールです。このモジュールは、デバイスのプログラム実行から独立してシリアルデータの送受信を実行するために必要なクロックジェネレータ、シフトレジスタ、データバッファを全て備えています。EUSART は SCI (Serial Communications Interface) とも呼ばれ、全二重非同期システムまたは半二重同期システムとして構成できます。全二重モードは、CRT 端末やパーソナルコンピュータ等の周辺システムとの通信に便利です。半二重同期モードは、A/D または D/A IC、シリアル EEPROM、その他のマイクロコントローラ等の周辺デバイスとの通信向けです。これらのデバイスは通常、baud レート生成用の内部クロックを持たないため、マスタ同期デバイスから外部クロック信号を供給する必要があります。

EUSART モジュールには以下の機能があります。

- 全二重非同期の送受信
- 2 文字の入力バッファ
- 1 文字の出力バッファ
- 文字長を 8 ビットまたは 9 ビットにプログラム可能
- アドレス検出 (9 ビットモード)
- 入力バッファ オーバーラン エラー検出
- 受信文字のフレーミング エラー検出
- 半二重同期マスタ
- 半二重同期スレーブ
- クロック極性をプログラム可能 (同期モード)
- スリープ動作

EUSART モジュールは以下の機能も実装しているため、LIN (Local Interconnect Network) バスシステムでの使用に理想的です。

- baud レートの自動検出および校正
- ブレーク受信による復帰
- 13 ビットのブレーク文字送信

図 33-1 と図 33-2 に、EUSART トランスミッタとレシーバのブロック図を示します。

EUSART トランスミッタ出力 (TX_out) は、TX/CK ピンに加え以下の内部モジュールでも利用できます。

- 構成可能なロジックセル (CLC)
- データ信号モジュレータ (DSM)

図 33-1: EUSART トランスミッタのブロック図

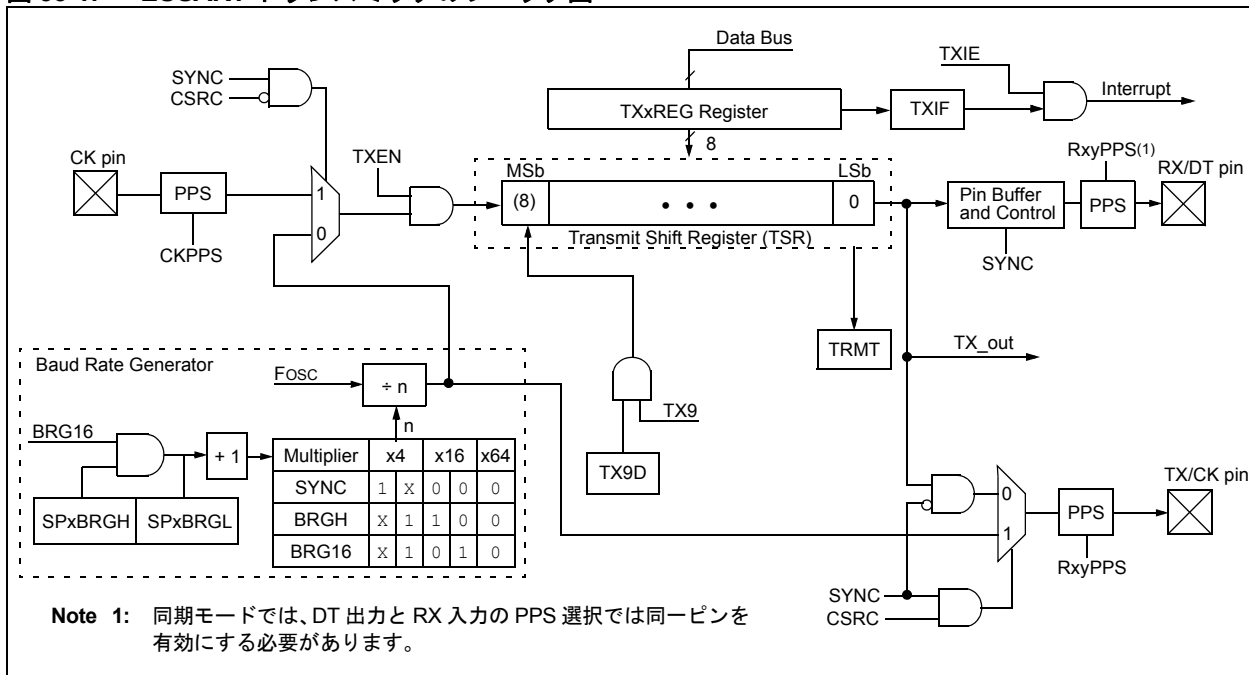
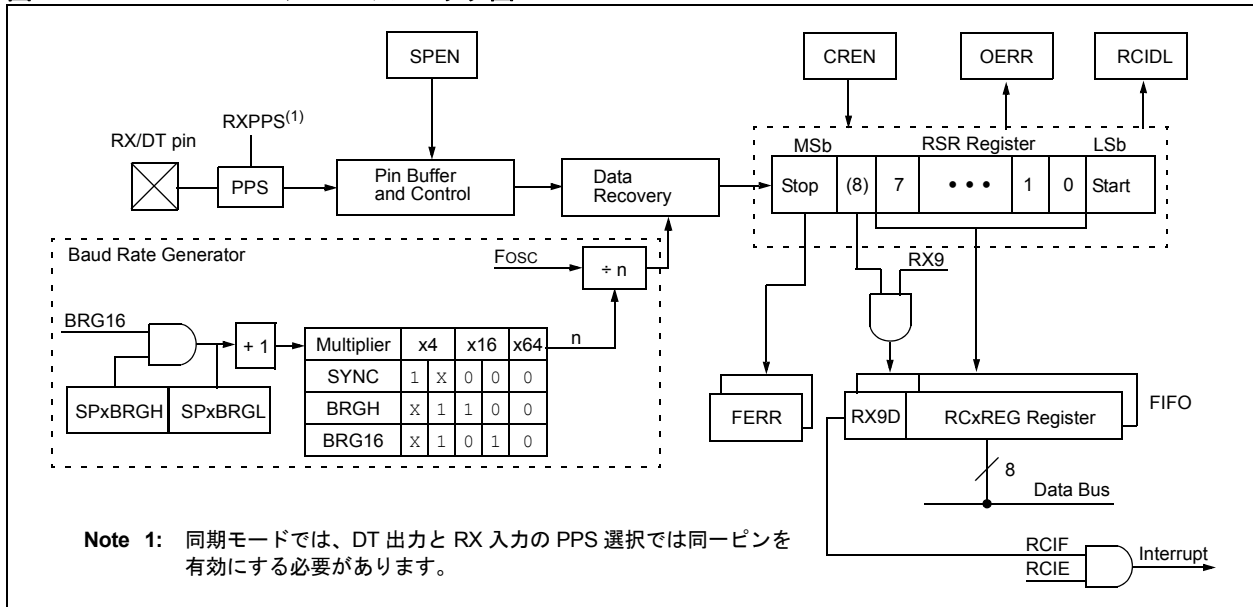


図 33-2: EUSART レシーバのブロック図



EUSART モジュールの動作は、以下の 3 つのレジスタで制御します。

- TXxSTA(送信ステータスおよび制御) レジスタ
- RCxSTA(受信ステータスおよび制御) レジスタ
- BAUDxCON (baud レート制御) レジスタ

これらのレジスタの詳細は、[レジスタ 33-1](#)、[レジスタ 33-2](#)、[レジスタ 33-3](#) をそれぞれ参照してください。

RX および CK 入力ピンは、それぞれ RXPPS および CKPPS レジスタで選択します。TX、CK、DT 出力ピンは、各ピンの RxyPPS レジスタで選択します。同期モードでは RX 入力と DT 出力が結合しているため、これら機能の両方に同じピンを選択するようにユーザーが注意する必要があります。データ方向のドライバは、EUSART 制御ロジックが制御します。

33.1 EUSART 非同期モード

EUSART は、標準の NRZ (Non-Return-to-Zero) 方式でデータを送受信します。NRZ は、データビット「1」を表す V_{OH} マーク状態と、データビット「0」を表す V_{OL} スペース状態の2つのレベルで実装されています。NRZ で同じ値のデータビットを連続して送信すると、出力レベルはそのビットのレベルのまま変化しません(1ビットを送信するたびにニュートラルレベルに戻るのではない)。NRZ 送信ポートのアイドル状態はマーク状態です。各送信文字は、1ビットのスタートビットの後に8ビットまたは9ビットのデータビットを送信し、最後は必ず1つまたは複数のストップビットで終了します。スタートビットは必ずスペースで、ストップビットは必ずマークです。最も一般的なデータフォーマットは8ビットです。1ビットの送信にかかる時間は、1/(baud レート)です。標準 baud レート周波数は、内蔵の専用8ビット/16ビット baud レートジェネレータを使ってシステムオシレータから生成します。baud レートの設定例は、表 33-5 を参照してください。

EUSART は LSb から順に送受信します。EUSART のトランスミッタとレシーバは独立して動作しますが、データフォーマットと baud レートは共通です。パリティはハードウェアレベルではサポートされていませんが、ソフトウェアで実装して9番目のデータビットとして格納できます。

33.1.1 EUSART 非同期トランスミッタ

図 33-1 に、EUSART トランスミッタのブロック図を示します。トランスミッタの中心となるのがシリアル送信シフトレジスタ (TSR) で、このレジスタにはソフトウェアから直接はアクセスできません。TSR は、送信バッファである TXxREG レジスタからデータを取得します。

33.1.1.1 トランスミッタの有効化

EUSART トランスミッタを非同期モードとして有効化するには、3つの制御ビットを以下のように設定します。

- TXEN = 1
- SYNC = 0
- SPEN = 1

その他の EUSART 制御ビットは全て既定値のままと仮定します。

TXxSTA レジスタの TXEN ビットをセットすると、EUSART のトランスミッタ回路が有効になります。TXxSTA レジスタの SYNC ビットをクリアすると、EUSART が非同期モードに設定されます。RCxSTA レジスタの SPEN ビットをセットすると EUSART が有効になり、TX/CK I/O ピンが自動的に出力に設定されます。TX/CK ピンをアナログ周辺モジュールと共用している場合、対応する ANSEL ビットをクリアしてアナログ I/O 機能を無効化する必要があります。

Note: TXEN イネーブルビットをセットすると、TXIF トランスミッタ割り込みフラグがセットされます。

33.1.1.2 送信データ

TXxREG レジスタに文字を書き込むと送信が開始します。これが最初の文字の場合、または TSR 内に以前の文字が全く残っていない場合、TXxREG のデータがただちに TSR レジスタに転送されます。TSR 内に以前の文字が一部でも残っている場合、その文字のストップビットが送信されるまで新しい文字データは TXxREG に保持されます。TXxREG に保留中の文字は、ストップビット送信直後の 1 T_{cy} の期間に TSR に転送されます。TXxREG から TSR にデータが転送されると、その直後にスタートビット、データビット、ストップビットのシーケンスの送信が開始します。

33.1.1.3 送信データの極性

送信データの極性は、BAUDxCON レジスタの SCKP ビットで設定します。このビットが既定値「0」の場合、送信アイドル/データビットは正論理です。SCKP ビットを「1」に設定すると送信データが反転し、送信アイドル/データビットは負論理になります。SCKP ビットで送信データの極性を制御できるのは、非同期モードの場合のみです。同期モードにおける SCKP ビットの機能は異なります。[セクション 33.5.1.2「クロック極性」](#)を参照してください。

33.1.1.4 送信割り込みフラグ

EUSART トランスミッタが有効で、かつ送信する文字が TXxREG に格納されていない場合は常に、PIR1 レジスタの TXIF 割り込みフラグビットがセットされます。つまり、TXIF ビットがクリアされるのは、TSR に何らかの文字がある時に新しい送信文字が TXxREG に格納された場合のみです。TXIF フラグビットは、TXxREG への書き込みの直後にはクリアされません。TXIF は、書き込みを実行してから2つ目の命令サイクルで有効になります。TXxREG への書き込み直後に TXIF をポーリングすると、正しい結果が得られません。TXIF ビットは読み出し専用で、ソフトウェアでセットまたはクリアする事はできません。

PIE1 レジスタの TXIE 割り込みイネーブルビットをセットすると、TXIF 割り込みが許可されます。しかし、TXxREG がエンプティになると TXIE イネーブルビットの状態に関係なく TXIF フラグビットがセットされます。

データ送信時に割り込みを使うには、引き続きデータを送信する場合のみ TXIE ビットをセットします。送信する最後の文字を TXxREG に書き込んだら、TXIE 割り込みイネーブルビットをクリアします。

33.1.1.5 TSR のステータス

TXxSTA レジスタの TRMT ビットが TSR レジスタの状態を示します。このビットは読み出し専用です。TRMT ビットは、TSR レジスタがエンプティになるとセットされ、TXxREG から TSR レジスタに文字が転送されるとクリアされます。TRMT ビットは、TSR レジスタから全てのビットがシフト出力されるまでクリアされたままです。このビットには割り込みロジックが接続されていないため、TSR のステータスを調べるにはこのビットをポーリングする必要があります。

Note: TSR レジスタはデータメモリに割り当てられていないため、ユーザからはアクセスできません。

33.1.1.6 9 ビット文字の送信

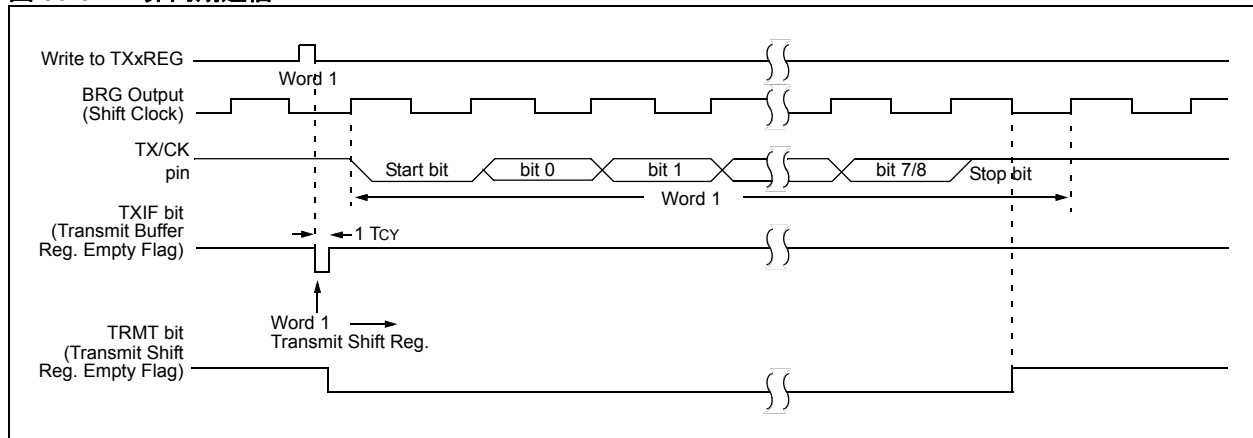
EUSART は 9 ビット文字の送信をサポートしています。TXxSTA レジスタの TX9 ビットをセットすると、EUSART は 1 文字の送信につき 9 ビットをシフト出力します。TXxSTA レジスタの TX9D ビットには、最上位 (9 ビット目) のデータビットを格納します。9 ビットのデータを送信する場合、まず TX9D ビットに書き込んでから TXxREG に下位 8 ビットを書き込む必要があります。TXxREG への書き込みが完了すると、ただちに 9 ビット全てのデータが TSR シフトレジスタに転送されます。

複数のレシーバを使う場合、特殊な 9 ビット アドレスモードが使えます。アドレスモードの詳細は、[セクション 33.1.2.7 「アドレス検出」](#) を参照してください。

33.1.1.7 非同期送信の実行手順:

1. 必要な baud レートに合わせて、SPxBRGH:SPxBRGL レジスタペアと BRGH および BRG16 の両ビットを初期化する ([セクション 33.4 「EUSART baud レート ジェネレータ \(BRG\)」](#) 参照)。
2. SYNC ビットをクリアし、SPEN ビットをセットして非同期シリアルポートを有効にする。
3. 9 ビット送信の場合、TX9 制御ビットをセットする。レシーバ側でアドレス検出を有効に設定している場合、9 ビット目をセットすると下位 8 ビットはアドレスと見なされる。
4. 送信データの極性を反転する場合、SCKP ビットをセットする。
5. TXEN 制御ビットをセットして、送信を有効にする。これにより、TXIF 割り込みビットがセットされる。
6. 割り込みが必要な場合、PIE1 レジスタの TXIE 割り込みイネーブルビットをセットする。INTCON レジスタの GIE および PEIE ビットもセットされていれば、割り込みがただちに発生する。
7. 9 ビット送信を選択した場合、9 ビット目を TX9D データビットに書き込む。
8. TXxREG レジスタに 8 ビットのデータを書き込むと、送信が開始する。

図 33-3: 非同期送信



PIC16(L)F1764/5/8/9

図 33-4: 非同期送信 (連続)

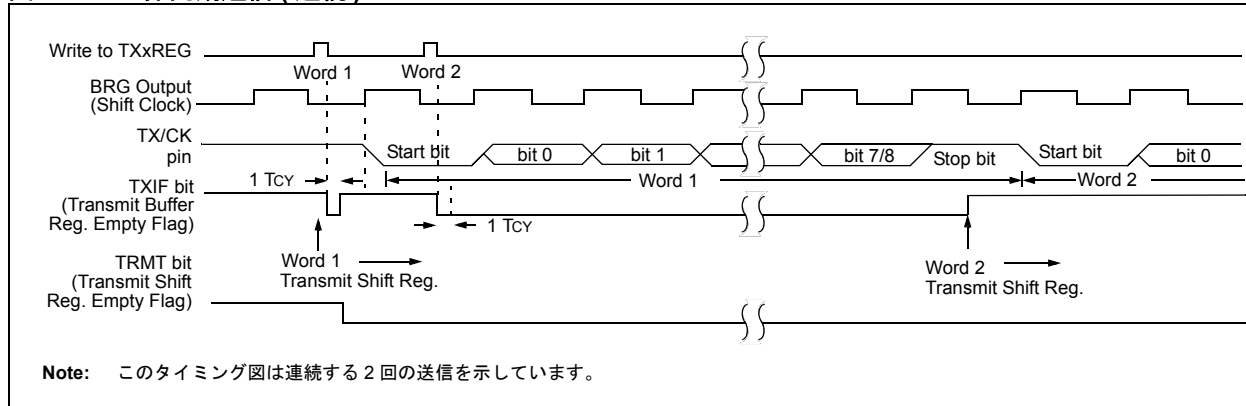


表 33-1: 非同期送信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RxyPPS	—	—	—	RxyPPS<4:0>					158, 160
SP1BRGL	BRG<7:0>								441*
SP1BRGH	BRG<15:8>								441*
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1REG	EUSART 送信データレジスタ								430*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は非同期送信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9のみです。

2: 未実装、「1」として読み出します。

33.1.2 EUSART 非同期レシーバ

通常、非同期モードは RS-232 システムで使います。図 33-2 に、レシーバのブロック図を示します。RX/DT ピンで受信したデータでデータ復元ブロックを駆動します。データ復元ブロックの実体は、baud レートの 16 倍の速度で動作する高速シフタです。一方、シリアル受信シフトレジスタ (RSR: Receive Shift Register) はビットレートで動作します。文字を構成する 8 ビットまたは 9 ビットの全てがシフト入力されると、ただちに 2 文字の FIFO (First-In-First-Out) メモリに転送されます。この FIFO バッファリングにより、ソフトウェアで EUSART レシーバのサービスを開始する前に、2 文字まで完全に受信し、3 文字目の受信を開始できます。FIFO と RSR レジスタには、ソフトウェアから直接はアクセスできません。受信データには、RCxREG レジスタを介してアクセスします。

33.1.2.1 レシーバの有効化

EUSART レシーバを非同期モードとして有効化するには、3 つの制御ビットを以下のように設定します。

- CREN = 1
- SYNC = 0
- SPEN = 1

その他の EUSART 制御ビットは全て既定値のままと仮定します。

RCxSTA レジスタの CREN ビットをセットすると、EUSART のレシーバ回路が有効になります。TXxSTA レジスタの SYNC ビットをクリアすると、EUSART が非同期モードに設定されます。RCxSTA レジスタの SPEN ビットをセットすると EUSART が有効になります。RX/DT I/O ピンを入力に設定するには、対応する TRIS ビットをセットする必要があります。

Note: RX/DT 機能がアナログピンに割り当てられている場合、対応する ANSEL ビットをクリアしないとレシーバは動作しません。

33.1.2.2 受信データ

レシーバのデータ復元回路は、最初のビットの立ち下がりエッジで文字の受信を開始します。最初のビット (スタートビット) は常にゼロです。データ復元回路はスタートビットの中央まで 1/2 ビットの期間をカウントし、そのビットがまだゼロかどうかをチェックします。ゼロでない場合、データ復元回路はエラーを生成せずに文字の受信を中止し、スタートビットの立ち下がりエッジの検出を再開します。スタートビットゼロの検証に成功した場合、データ復元回路は次のビットの中央までさらに 1 ビットの期間をカウントします。次にこのビットを多数決回路でサンプリングし、得られた結果 ('0' または '1') を RSR にシフト入力します。以後、この動作を繰り返して全てのデータビットをサンプリングし、RSR にシフト入力します。最後の 1 ビットの期間を計測して、そのレベルをサンプリングします。これはストップビットで、常に '1' です。ストップビットの位置でデータ復元回路が '0' をサンプリングした場合、この文字に対してフレーミングエラーがセットされます。それ以外の場合、この文字のフレーミングエラーはクリアされます。フレーミングエラーの詳細は、[セクション 33.1.2.4 「受信フレーミングエラー」](#) を参照してください。

全てのデータビットとストップビットを受信すると、RSR 内の文字はただちに EUSART 受信 FIFO に転送され、PIR1 レジスタの RCIF 割り込みフラグビットがセットされます。RCxREG レジスタを読み出すと、FIFO から先頭の文字が出力されます。

Note: 受信 FIFO がオーバーランした場合、オーバーラン条件が解消されるまで、それ以降の文字は受信されません。オーバーランエラーの詳細は、[セクション 33.1.2.5 「受信オーバーランエラー」](#) を参照してください。

33.1.2.3 受信割り込み

EUSART レシーバが有効で、受信 FIFO 内に読み出していない文字がある場合は常に、PIR1 レジスタの RCIF 割り込みフラグビットがセットされます。RCIF 割り込みフラグビットは読み出し専用で、ソフトウェアでセットまたはクリアする事はできません。

以下の全ビットをセットすると RCIF 割り込みが有効になります。

- PIE1 レジスタの RCIE 割り込みイネーブルビット
- INTCON レジスタの PEIE 周辺モジュール割り込みイネーブルビット
- INTCON レジスタの GIE グローバル割り込みイネーブルビット

FIFO 内に読み出していない文字が残っている場合、これらの割り込みイネーブルビットの状態に関係なく RCIF 割り込みフラグビットがセットされます。

33.1.2.4 受信フレーミングエラー

受信 FIFO バッファ内の各文字には、対応するフレーミングエラーステータスビットがあります。フレーミングエラーは、ストップビットが本来の位置で正しく検出されなかった場合に発生します。フレーミングエラーステータスは、RCxSTA レジスタの FERR ビットで調べる事ができます。FERR ビットは、受信 FIFO の先頭にある、読み出されていない文字のステータスを示します。従って、RCxREG を読み出す前に FERR ビットを読み出す必要があります。

FERR ビットは読み出し専用で、受信 FIFO の先頭にある、読み出されていない文字にのみ適用されます。フレーミングエラーが発生しても (FERR = 1)、後続文字の受信を妨げません。FERR ビットはクリアする必要があります。FIFO バッファから次の文字を読み出すと、FIFO は次の文字に進み、FERR ビットはその文字のフレーミングエラーステータスを格納します。

RCxSTA レジスタの SPEN ビットをクリアして EUSART をリセットすると、FERR ビットは強制的にクリアされます。RCxSTA レジスタの CREN ビットをクリアしても FERR ビットには影響しません。フレーミングエラー自体は、割り込みを生成しません。

Note: 受信 FIFO 内の全ての受信文字にフレーミングエラーがある場合、RCxREG を繰り返し読み出しても FERR ビットはクリアされません。

33.1.2.5 受信オーバーランエラー

受信 FIFO バッファには2文字まで格納できます。FIFO にアクセスする前に3文字目のデータを最後まで受信すると、オーバーランエラーが発生します。オーバーランエラーになると、RCxSTA レジスタの OERR ビットがセットされます。この場合、既に FIFO バッファ内にある文字は読み出す事ができますが、それ以降の文字はオーバーランエラーが解消されるまで受信できません。オーバーランエラーを解消するには、RCxSTA レジスタの CREN ビットをクリアするか、RCxSTA レジスタの SPEN ビットをクリアして EUSART をリセットする必要があります。

33.1.2.6 9ビット文字の受信

EUSART は9ビット文字の受信をサポートしています。RCxSTA レジスタの RX9 ビットがセットされている場合、EUSART は1文字の受信につき9ビットを RSR にシフト入力します。RCxSTA レジスタの RX9D ビットには、受信 FIFO の先頭の読み出していない文字の9ビット目(最上位データビット)が格納されます。受信 FIFO バッファから9ビットデータを読み出す場合、RCxREG の下位8ビットより前に RX9D データビットを読み出す必要があります。

33.1.2.7 アドレス検出

RS-485 システムのように1つの伝送ラインを複数のレシーバが共用している場合、特別なアドレス検出モードを利用できます。RCxSTA レジスタの ADDEN ビットをセットするとアドレス検出が有効になります。

アドレス検出モードを利用するには、9ビット文字を受信する必要があります。アドレス検出を有効化すると、9番目のデータビットがセットされた文字のみが受信 FIFO バッファに転送され、これによって RCIF 割り込みビットがセットされます。これ以外の文字は全て無視されます。

アドレス文字を受信したら、そのアドレスが自分のアドレスと一致するかどうかをユーザソフトウェアで判定します。アドレスが一致したら、次のストップビットが発生する前にユーザソフトウェアで ADDEN ビットをクリアしてアドレス検出を無効にする必要があります。使っているメッセージプロトコルに基づいてメッセージの終了を検出したら、ユーザソフトウェアで ADDEN ビットをセットして、レシーバを再びアドレス検出モードに戻します。

33.1.2.8 非同期受信の実行手順

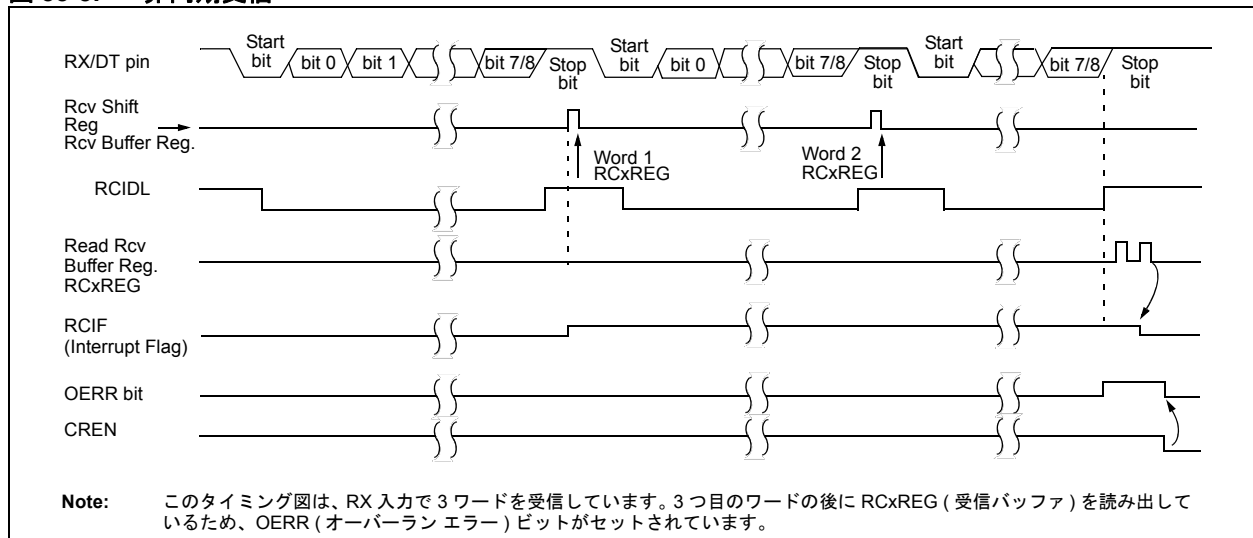
1. 必要な baud レートに合わせて、SPxBRGH:SPxBRGL レジスタペアと BRGH および BRG16 の両ビットを初期化する ([セクション 33.4 「EUSART baud レート ジェネレータ \(BRG\)」](#) 参照)。
2. RXピンのANSELビットをクリアする(必要な場合)。
3. SPEN ビットをセットしてシリアルポートを有効にする。SYNC ビットをクリアして、非同期モードに設定する。
4. 割り込みが必要な場合、PIE1 レジスタの RCIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
5. 9ビット受信を使う場合、RX9ビットをセットする。
6. CREN ビットをセットして、受信を有効にする。
7. RSRから受信バッファに文字が転送されると、RCIF 割り込みフラグビットがセットされる。RCIE 割り込みイネーブルビットもセットしている場合、割り込みが発生する。
8. RCxSTA レジスタを読み出してエラーフラグを取得する。9 ビットデータ受信を有効にしている場合、9 ビット目のデータビットも読み出す。
9. RCxREG レジスタを読み出し、受信バッファから受信済みデータの下位 8 ビットを取得する。
10. オーバーランが発生した場合、CREN レシーバイネーブルビットをクリアしてOERRフラグをクリアする。

33.1.2.9 9 ビットアドレス検出モードの実行手順

通常、このモードは RS-485 システムで使います。非同期受信でアドレス検出を有効にするには、以下の手順を実行します。

1. 必要な baud レートに合わせて、SPxBRGH:SPxBRGL レジスタペアと BRGH および BRG16 の両ビットを初期化する ([セクション 33.4 「EUSART baud レート ジェネレータ \(BRG\)」](#) 参照)。
2. RXピンのANSELビットをクリアする(必要な場合)。
3. SPEN ビットをセットしてシリアルポートを有効にする。SYNC ビットをクリアして、非同期モードに設定する。
4. 割り込みが必要な場合、PIE1 レジスタの RCIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
5. RX9ビットをセットして9ビット受信を有効にする。
6. ADDEN ビットをセットしてアドレス検出を有効にする。
7. CREN ビットをセットして、受信を有効にする。
8. 9 ビット目をセットされた文字が RSR から受信バッファに転送されると、RCIF 割り込みフラグビットがセットされる。RCIE 割り込みイネーブルビットもセットしている場合、割り込みが発生する。
9. RCxSTA レジスタを読み出してエラーフラグを取得する。9 ビット目のデータビットは常にセットされる。
10. RCxREG レジスタを読み出し、受信バッファから受信済みデータの下位 8 ビットを取得する。これがデバイスのアドレスかどうかをソフトウェアで判定する。
11. オーバーランが発生した場合、CREN レシーバイネーブルビットをクリアしてOERRフラグをクリアする。
12. デバイスのアドレスと一致した場合、ADDENビットをクリアして全ての受信データを受信バッファに取り込み、割り込みを生成する。

図 33-5: 非同期受信



PIC16(L)F1764/5/8/9

表 33-2: 非同期受信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1REG	EUSART 受信データレジスタ								433*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RxyPPS	—	—	—	RxyPPS<4:0>					158
SP1BRGL	BRG<7:0>								441
SP1BRGH	BRG<15:8>								441
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1STA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は非同期受信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9のみです。

2: 未実装、「1」として読み出します。

33.2 非同期モードのクロック精度

内部オシレータ ブロックの出力 (INTOSC) は工場では校正済みです。しかし、INTOSC の周波数は VDD または温度の変化によってドリフトする事があり、これによって非同期 baud レートが直接影響を受けます。baud レートクロックを調整するには 2 つの方法がありますが、どちらの場合も何らかの参照クロック源が必要です。

1 つは OSCTUNE レジスタを使って INTOSC 出力を調整する方法で、通常はこちらの方法を推奨します。OSCTUNE レジスタの値を調整すると、システムクロック源を微調整できます。詳細は [セクション 5.2.2.3 「内部オシレータの周波数調整」](#) を参照してください。

もう 1 つは、baud レート ジェネレータの値を調整する方法です。これは、自動 baud レート検出機能によって自動的に実行できます ([セクション 33.4.1 「自動 baud レート検出」](#) 参照)。ただし、baud レート ジェネレータを調整して周辺クロック周波数の経時変化を補償する場合、分解能が不十分な事があります。

PIC16(L)F1764/5/8/9

33.3 レジスタ定義 : EUSART 制御

レジスタ 33-1: TX1STA: 送信ステータスおよび制御レジスタ

R/W-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **CSRC**: クロック源選択ビット
 非同期モード:
 ドントケア
 同期モード:
 1 = マスタモード (クロックを内部の BRG から生成する)
 0 = スレーブモード (外部クロック源を使う)
- bit 6 **TX9**: 9 ビット送信イネーブルビット
 1 = 9 ビット送信を選択する
 0 = 8 ビット送信を選択する
- bit 5 **TXEN**: 送信イネーブルビット ⁽¹⁾
 1 = 送信を有効にする
 0 = 送信を無効にする
- bit 4 **SYNC**: EUSART モード選択ビット
 1 = 同期モード
 0 = 非同期モード
- bit 3 **SENDB**: ブレーク文字送信ビット
 非同期モード:
 1 = 次の送信で同期ブレークを送信する (完了時にハードウェアでクリアされる)
 0 = 同期ブレークの送信が完了している
 同期モード:
 ドントケア
- bit 2 **BRGH**: 高 baud レート選択ビット
 非同期モード:
 1 = 高速
 0 = 低速
 同期モード:
 このモードでは使わない
- bit 1 **TRMT**: 送信シフトレジスタ ステータスビット
 1 = TSR エンプティ
 0 = TSR フル
- bit 0 **TX9D**: 送信データの 9 番目のビット
 アドレス / データビットまたはパリティビットとして使用可能

Note 1: 同期モードの場合、SREN/CREN の設定が TXEN に優先します。

レジスタ 33-2: RC1STA: 受信ステータスおよび制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **SPEN:** シリアルポート イネーブルビット
1 = シリアルポートを有効にする
0 = シリアルポートを無効にする (リセットに保持)
- bit 6 **RX9:** 9 ビット受信イネーブルビット
1 = 9 ビット受信を選択する
0 = 8 ビット受信を選択する
- bit 5 **SREN:** シングル受信イネーブルビット
非同期モード:
ドントケア
同期モード - マスタ:
1 = シングル受信を有効にする
0 = シングル受信を無効にする
このビットは受信完了後にクリアされます。
同期モード - スレーブ:
ドントケア
- bit 4 **CREN:** 連続受信イネーブルビット
非同期モード:
1 = レシーバを有効にする
0 = レシーバを無効にする
同期モード:
1 = イネーブルビット CREN がクリアされるまで連続受信を有効にする (CREN の設定が SREN に優先される)
0 = 連続受信を無効にする
- bit 3 **ADDEN:** アドレス検出イネーブルビット
非同期モード 9 ビット (RX9 = 1):
1 = アドレス検出、割り込みを有効にする。RSR<8> がセットされたら受信バッファに書き込む
0 = アドレス検出を無効にする。全てのバイトが受信され、9 ビット目はパリティビットとして使える
非同期モード 8 ビット (RX9 = 0):
ドントケア
- bit 2 **FERR:** フレーミング エラービット
1 = フレーミング エラーあり (RCxREG レジスタを読み出すと更新され、次の有効なバイトを受信する)
0 = フレーミング エラーは発生していない
- bit 1 **OERR:** オーバーラン エラービット
1 = オーバーラン エラーあり (CREN ビットをクリアするとクリアされる)
0 = オーバーラン エラーは発生していない
- bit 0 **RX9D:** 受信データの 9 番目のビット
アドレス / データビットまたはパリティビットとして使える。ユーザ ファームウェアにより求める必要がある

PIC16(L)F1764/5/8/9

レジスタ 33-3: BAUD1CON: baud レート制御レジスタ

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
u = ビットは不変 x = ビットは未知 -n/n = POR 時と BOR 時の値 / その他の全てのリセット時の値
「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **ABDOVF:** 自動 baud レート検出オーバーフロー ビット
非同期モード:
1 = 自動 baud レートタイマがオーバーフローした
0 = 自動 baud レートタイマはオーバーフローしていない
同期モード:
ドントケア
- bit 6 **RCIDL:** 受信アイドルフラグ ビット
非同期モード:
1 = レシーバはアイドル中
0 = スタートビットが受信され、レシーバが受信中等である
同期モード:
ドントケア
- bit 5 **未実装:** 「0」として読み出し
- bit 4 **SCKP:** 同期クロック極性選択ビット
非同期モード:
1 = TX/CK ピンへ反転データを送信する
0 = TX/CK ピンへ非反転データを送信する
同期モード:
1 = データはクロックの立ち上がりエッジを参照する
0 = データはクロックの立ち下がりエッジを参照する
- bit 3 **BRG16:** 16 ビット baud レート ジェネレータ ビット
1 = 16 ビット baud レート ジェネレータを使う
0 = 8 ビット baud レート ジェネレータを使う
- bit 2 **未実装:** 「0」として読み出し
- bit 1 **WUE:** 復帰イネーブルビット
非同期モード:
1 = レシーバは立ち下がりエッジの待機中である。文字は受信されず、エッジ検出後 RCIF ビットがセットされる。RCIF がセットされると WUE は自動的にクリアされる
0 = レシーバは通常動作中である
同期モード:
ドントケア
- bit 0 **ABDEN:** 自動 baud レート検出イネーブルビット
非同期モード:
1 = 自動 baud レート検出モードを有効にする (自動 baud レート検出が完了するとクリアされる)
0 = 自動 baud レート検出モードを無効にする
同期モード:
ドントケア

33.4 EUSART baud レート ジェネレータ (BRG)

baud レート ジェネレータ (BRG) は、EUSART の非同期および同期モードの両方をサポートするための専用の 8 ビットまたは 16 ビットタイマです。既定値では、BRG は 8 ビットモードで動作します。BAUDxCON レジスタの BRG16 ビットをセットすると、16 ビットモードを選択できます。

baud レートタイマはフリーランニングで、その周期は SPxBRGH:SPxBRGL レジスタペアで決定します。非同期モードでは、TxSTA レジスタの BRGH ビットと BAUDxCON レジスタの BRG16 ビットの両方で baud レート周期の乗数が決まります。同期モードでは、BRGH ビットは無視されます。

表 33-3 に、baud レートの計算式を示します。例 33-1 に、baud レートおよび baud レートエラーの計算例を示します。

参考までに、各種非同期モードにおける代表的な baud レートと baud レート誤差の計算値を表 33-5 にまとめます。高 baud レート (BRGH = 1) または 16 ビット BRG (BRG16 = 1) を使うと、baud レート誤差の低減に効果があります。16 ビット BRG モードを使うと、オシレータ周波数が高い場合でも低速の baud レートが得られます。

SPxBRGH:SPxBRGL レジスタペアに新しい値を書き込むと、BRG タイマがリセット (クリア) されます。これにより、BRG はタイマ オーバーフローを待たずに新しい baud レートを出力します。

レシーバの受信動作中にシステムクロックを変更すると、レシーバエラーまたはデータの損失が発生します。この問題を防ぐには、システムクロックを変更する前に RCIDL ビットの状態を調べて、受信動作がアイドル状態である事を確認してください。

例 33-1: BAUD レート誤差の計算

Fosc = 16 MHz、必要な baud レートが 9600、非同期モード、8 ビット BRG のデバイスの場合：

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

SPxBRGH:SPxBRGL について解くと：

$$X = \frac{F_{OSC}}{64 \cdot \text{Desired Baud Rate}} - 1$$

$$\begin{aligned} &= \frac{16000000}{64 \cdot 9600} - 1 \\ &= [25.042] = 25 \end{aligned}$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

PIC16(L)F1764/5/8/9

表 33-3: baud レートの計算式

コンフィグレーションビット			BRG/EUSART モード	baud レートの計算式
SYNC	BRG16	BRGH		
0	0	0	8 ビット / 非同期	$F_{osc}/[64 (n+1)]$
0	0	1	8 ビット / 非同期	$F_{osc}/[16 (n+1)]$
0	1	0	16 ビット / 非同期	
0	1	1	16 ビット / 非同期	$F_{osc}/[4 (n+1)]$
1	0	x	8 ビット / 同期	
1	1	x	16 ビット / 同期	

凡例: x = ドントケア、n = SPxBRGH:SPxBRGL レジスタペアの値

表 33-4: baud レート ジェネレータ関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
SP1BRGL	BRG<7:0>								441
SP1BRGH	BRG<15:8>								441
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は baud レート ジェネレータでは使いません。

* 本文中でレジスタを説明しているページです。

表 33-5: 各種非同期モードにおける baud レート

baud レート	SYNC = 0、BRGH = 0、BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

baud レート	SYNC = 0、BRGH = 0、BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

baud レート	SYNC = 0、BRGH = 1、BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.64k	2.12	16	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

PIC16(L)F1764/5/8/9

表 33-5: 各種非同期モードにおける baud レート (続き)

baud レート	SYNC = 0、BRGH = 1、BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

baud レート	SYNC = 0、BRGH = 0、BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

baud レート	SYNC = 0、BRGH = 0、BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

PIC16(L)F1764/5/8/9

表 33-5: 各種非同期モードにおける baud レート (続き)

baud レート	SYNC = 0、BRGH = 1、BRG16 = 1 または SYNC = 1、BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

baud レート	SYNC = 0、BRGH = 1、BRG16 = 1 または SYNC = 1、BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)	実際の レート	% 誤差	SPxBRG の値 (10 進数)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

PIC16(L)F1764/5/8/9

33.4.1 自動 baud レート検出

EUSART モジュールは baud レートの自動検出 / 校正をサポートしています。

自動 baud レート検出 (ABD) モードでは、BRG へのクロックが逆方向になります。BRG が入力 RX 信号にクロックを供給するのではなく、RX 信号が BRG のタイミングを生成します。baud レート ジェネレータを使って、LIN バスの同期文字である 55h (ASCII「U」) の受信周期を計測します。この文字は、ストップビットのエッジを含め 5 つの立ち上がりエッジがあるのが特徴です。

BAUDxCON レジスタの ABDEN ビットをセットすると、自動 baud レート校正シーケンスが開始します。ABD シーケンスの実行中、EUSART ステートマシンはアイドルのまま保持されます。スタートビット直後の受信ラインの最初の立ち上がりエッジで、SPxBRG が BRG カウンタクロックを使ってカウントを開始します (図 33-6 参照)。5 つ目の立ち上がりエッジは、8 番目のビット周期の最後に RX ピンで発生します。この時点で、SPxBRGH:SPxBRGL レジスタペア内は正しい BRG 周期を累計した値を格納しており、ABDEN ビットは自動的にクリアされ、RCIF 割り込みフラグがセットされます。RCIF 割り込みをクリアするには、RCxREG レジスタの値を読み出す必要があります。RCxREG の内容は破棄します。SPxBRGH レジスタを使わないモードで校正する場合、SPxBRGH レジスタの 00h をユーザがチェックすれば、SPxBRGL レジスタがオーバフローしていない事を確認できます。

BRG の自動 baud レートクロックは、BRG16 ビットと BRGH ビットによって決まります (表 33-6 参照)。ABD モードでは、SPxBRGH および SPxBRGL レジスタを BRG16 ビットの設定に関係なく 16 ビットカウンタとして使います。baud レート周期の校正時、SPxBRGH および SPxBRGL レジスタは BRG ベース

クロックの 1/8 のレートで動作します。こうして得られたバイトの計測値は、フルスピードでのクロック時の平均ビット時間です。

Note 1: ABDEN ビットと WUE ビットを両方セットした場合、自動 baud レート検出はブレーク文字直後のバイトで実行されます (セクション 33.4.3「ブレークによる自動復帰」参照)。

2: 入力文字の baud レートが、選択した BRG クロック源のレンジ内であるかどうかの判断は、ユーザに委ねられています。オンレータ周波数と EUSART の baud レートの組み合わせによっては、使えないものもあります。

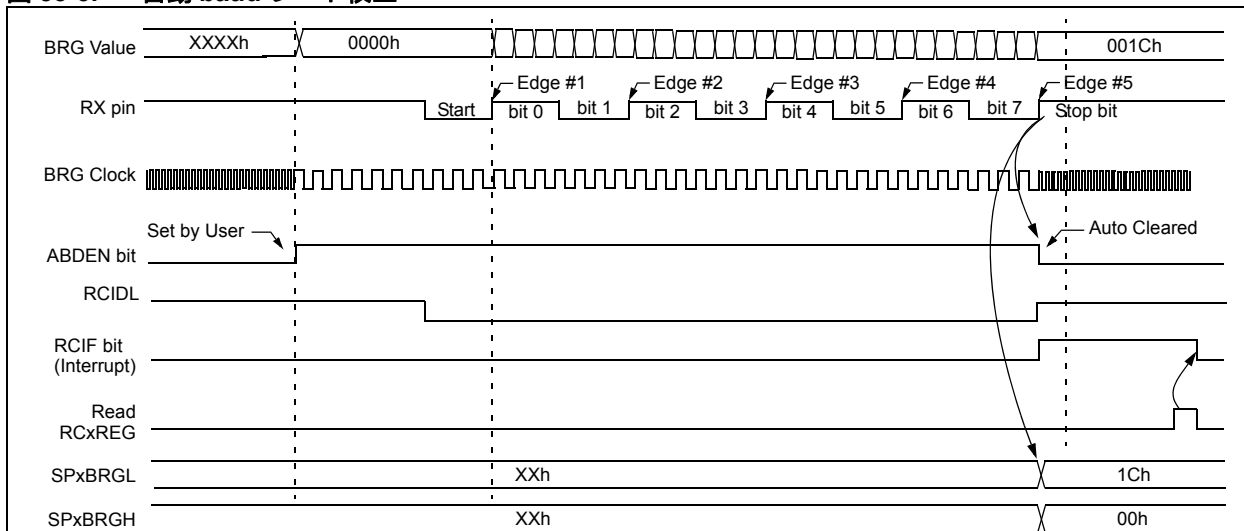
3: 自動 baud レートプロセスでは、自動 baud レートカウンタが 1 からカウントを開始します。精度を重視する場合、自動 baud レートシーケンスの完了時に SPxBRGH:SPxBRGL レジスタペアから 1 を引きます。

表 33-6: BRG カウンタのクロックレート

BRG16	BRGH	BRG ベース クロック	BRG ABD クロック
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

Note: ABD シーケンスでは、SPxBRGL および SPxBRGH レジスタを BRG16 ビットの設定に関係なく 16 ビットカウンタとして使います。

図 33-6: 自動 baud レート校正



Note 1: ABD シーケンスを実行するには、EUSART モジュールを非同期モードに設定する必要があります。

33.4.2 自動 baud レートのオーバーフロー

自動 baud レート検出時、RX ピンで 5 つ目の立ち上がりエッジを検出する前に baud レートカウンタがオーバーフローすると、BAUDxCON レジスタの ABDOVF ビットがセットされます。ABDOVF ビットは、カウンタの値が SPxBRGH:SPxBRGL レジスタペアの 16 ビットに格納できる最大カウントを超えた事を示します。オーバーフロー条件になると RCIF フラグがセットされます。カウンタは RX ピンで 5 つ目の立ち上がりエッジを検出するまでカウントを継続します。RCIDL ビットは 5 つ目の立ち上がりエッジでセットされるまで偽 (「0」) を維持します。5 つ目の立ち上がりエッジ前にオーバーフローが発生し RCREG を読み出した場合、RCIF は 5 つ目の立ち上がりエッジで再度セットされます。

オーバーフロー条件を解消する前に自動 baud レートプロセスを終了すると、同期文字の 5 つ目の立ち上がりエッジの検出が正しく行われません。ABDEN ビットがクリアされた時に同期文字の立ち下がりエッジが発生していない場合、立ち下がりエッジがスタートビットとして誤検出される事になります。オーバーフロー条件を解消するには、以下のステップに従う事を推奨します。

1. RCREG を読み出し RCIF をクリアする。
2. RCIDL がゼロの場合、RCIF がセットされるのを待ってからステップ 1 を繰り返す。
3. ABDOVF ビットをクリアする。

33.4.3 ブレークによる自動復帰

スリープモード中は、EUSART へのクロックは全て停止されます。このため、baud レート ジェネレータは非アクティブ状態で、文字を正しく受信できません。自動復帰機能を利用すると、RX/DT ラインのアクティビティによってコントローラを復帰できます。この機能は、非同期モードでのみ使う事ができます。

自動復帰機能を有効にするには、BAUDxCON レジスタの WUE ビットをセットします。このビットをセットすると RX/DT での通常受信シーケンスは無効となり、EUSART はアイドル状態のまま、CPU のモードとは関係なく復帰イベントの発生を監視します。RX/DT ラインが High から Low に遷移すると復帰イベントとなります (これは、LIN プロトコルの同期ブレークまたは復帰信号文字の開始に相当します)。

復帰イベントが発生すると同時に、EUSART モジュールは RCIF 割り込みを生成します。この割り込みは、CPU が通常の動作モードの場合は Q クロックに同期して発生し (図 33-7)、デバイスがスリープ中は非同期に発生します (図 33-8)。RCxREG レジスタを読み出すと、割り込み条件がクリアされます。

ブレークの最後に RX ラインが Low から High に遷移すると、WUE ビットは自動的にクリアされます。これにより、ユーザはブレークイベントの終了を知ることができます。この時点で EUSART モジュールはアイドルになり、次の文字の受信を待ちます。

33.4.3.1 注意事項

ブレーク文字

復帰イベント時の文字エラーまたは文字の断片化を防止するため、復帰文字は全てゼロにする必要があります。

復帰を有効にすると、データストリームの Low 時間とは関係なくこの機能が働きます。WUE ビットをセットした場合にゼロ以外の有効な文字を受信すると、スタートビットから最初の立ち上がりエッジまでの Low 時間が復帰イベントとして解釈されます。文字の残りのビットは断片化した文字として受信され、後続の文字がフレーミング エラーまたはオーバーラン エラーとなる事があります。

従って、最初に送信する文字は全て「0」である必要があります。この状態を 10 ビット以上の期間持続する必要があります。LIN バスでは 13 ビットの期間を推奨しますが、標準の RS-232 デバイスでは任意のビット期間でかまいません。

オシレータの起動時間

LP、XT、HS/PLL モード等、特に起動時間の長いオシレータを使う応用回路では、オシレータの起動時間を考慮する必要があります。十分な長さの同期ブレーク文字 (または復帰信号) を送信したら、選択したオシレータが起動するまで十分な間隔を置いてから EUSART を適切に初期化します。

WUE ビット

復帰イベントが発生すると、RCIF ビットのセットによって受信割り込みが発生します。WUE ビットは、RX/DT の立ち上がりエッジでハードウェアによってクリアされます。その後、RCxREG レジスタを読み出してその内容を破棄すると、割り込み条件はソフトウェアによってクリアされます。

実際のデータが失われないようにするには、WUE ビットをセットする前に RCIDL ビットを調べて、受信動作中でない事を確認します。受信動作中でなければ、スリープに移行する直前に WUE ビットをセットできます。

PIC16(L)F1764/5/8/9

図 33-7: 通常動作時の自動復帰ビット (WUE) のタイミング

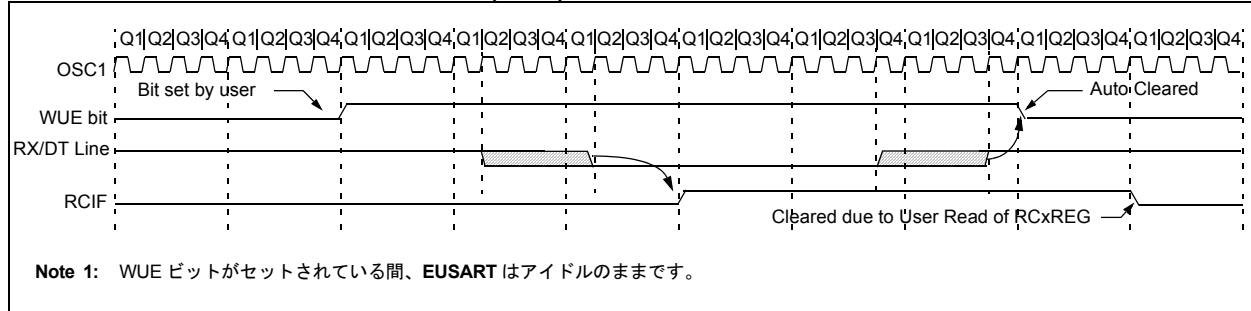
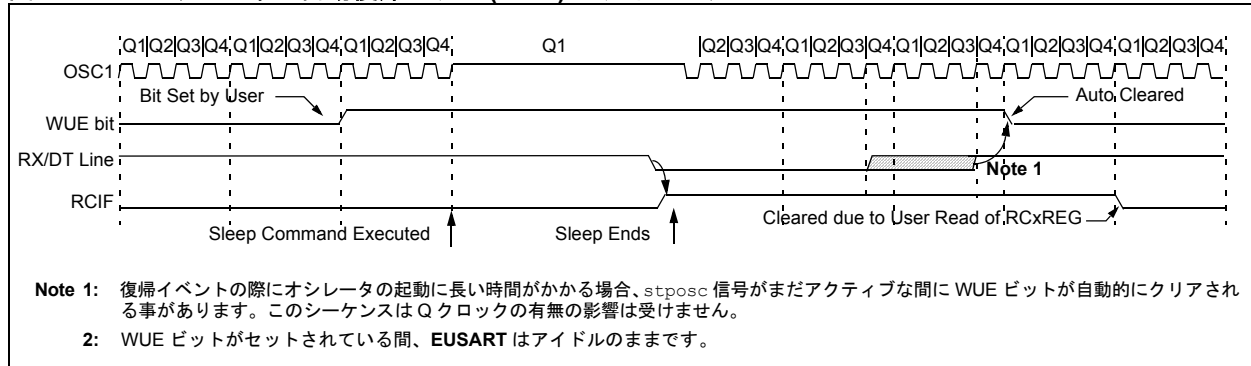


図 33-8: スリープ中の自動復帰ビット (WUE) のタイミング



33.4.4 ブレーク文字のシーケンス

EUSART モジュールには、LIN バス規格で必要とされる特殊なブレーク文字シーケンスを送信する機能があります。ブレーク文字は、スタートビット、12 ビットの「0」、ストップビットの順で構成されます。

ブレーク文字を送信するには、まず TXxSTA レジスタの SENDB および TXEN ビットをセットします。次に、TXxREG に書き込みを実行するとブレーク文字の送信が開始します。TXxREG に書き込んだデータの値は無視され、全て「0」として送信されます。

対応するストップビットの送信後、SENDB ビットはハードウェアによって自動的にリセットされます。このため、ブレーク文字の次に送信するバイト（通常、LIN 規格の場合は同期文字）を送信 FIFO にあらかじめ読み込んでおく事ができます。

TXxSTA レジスタの TRMT ビットは、通常の実送時と同様に、送信動作がアクティブかアイドルかを示します。図 33-9 に、ブレーク文字送信シーケンスのタイミングを示します。

33.4.4.1 ブレークおよび同期文字の送信シーケンス

以下のシーケンスにより、メッセージ フレームヘッダ（ブレーク文字 + 自動 baud レート同期バイト）が開始します。このシーケンスは、LIN バスマスタの代表的なものです。

1. EUSART を目的のモードに設定する。
2. TXEN および SENDB ビットをセットしてブレークシーケンスを有効にする。
3. TXxREG にダミー文字を書き込んで送信を開始する（書き込んだ値は無視される）。
4. TXxREG に「55h」を書き込んで送信 FIFO バッファに同期文字を格納する。
5. ブレークの送信が完了したら、SENDB ビットがハードウェアによってリセットされ、同期文字が送信される。

TXxREG がエンプティになった事が TXIF によって示されたら、次のデータバイトを TXxREG に書き込む事ができます。

33.4.5 ブレーク文字の受信

EUSART モジュールには、ブレーク文字の受信方法が 2 通りあります。

1 つは、RCxSTA レジスタの FERR ビットと RCxREG が示す受信データを使ってブレーク文字を検出する方法です。baud レート ジェネレータは、予測 baud レートに初期化されているものと仮定します。

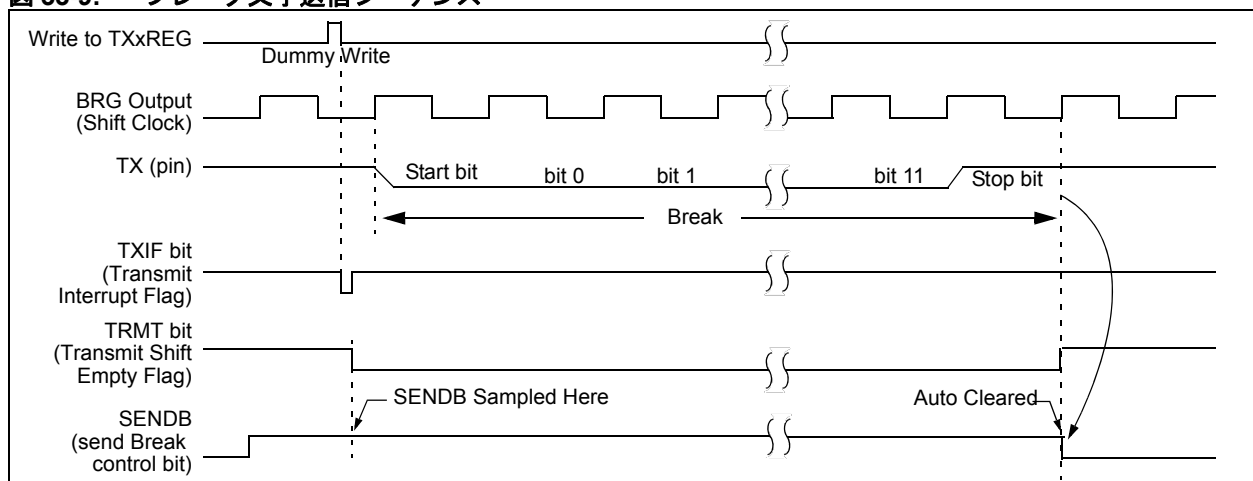
以下の条件が揃うと、ブレーク文字が受信されたと判断できます。

- RCIF = 1
- FERR = 1
- RCxREG = 00h

もう 1 つは、自動復帰機能（[セクション 33.4.3「ブレークによる自動復帰」](#)参照）を使う方法です。自動復帰機能を有効にすると、EUSART は RX/DT の次の 2 つの遷移をサンプリングし、RCIF 割り込みを生成して、次のデータバイトとその後の割り込みを受信します。

通常は、ブレーク文字の後に自動 baud レート検出機能を有効にします。どちらの方法でも、ユーザは EUSART をスリープに移行させる前に BAUDxCON レジスタの ABDEN ビットをセットできます。

図 33-9: ブレーク文字送信シーケンス



33.5 EUSART 同期モード

一般に、同期シリアル通信は1つのマスタと1つまたは複数のスレーブで構成されるシステムで使います。マスタデバイスは baud レート生成に必要な回路を持ち、システム内の全てのデバイスにクロックを供給します。スレーブデバイスは内部クロック生成回路を持たず、マスタから供給されるクロックを使います。

同期モードでは、双方向データラインとクロックラインの2本の信号線を使います。スレーブは、マスタから供給される外部クロックを使って送信 / 受信シフトレジスタそれぞれにシリアルデータをシフト出力 / 入力します。データラインは双方向であるため、同期動作は半二重のみです。半二重では、マスタとスレーブのどちらのデバイスからもデータを送受信できますが、両方が同時に送信する事はできません。EUSART モジュールはマスタデバイスとしてもスレーブデバイスとしても動作が可能です。

同期送信では、スタートビットとストップビットを使いません。

33.5.1 同期マスタモード

EUSART モジュールを同期マスタモードに設定するには、以下のようにビットを設定します。

- SYNC = 1
- CSRC = 1
- SREN = 0(送信の場合)、SREN = 1(受信の場合)
- CREN = 0(送信の場合)、CREN = 1(受信の場合)
- SPEN = 1

TXxSTA レジスタの SYNC ビットをセットすると、デバイスが同期モードに設定されます。TXxSTA レジスタの CSRC ビットをセットすると、デバイスがマスタに設定されます。RCxSTA レジスタの SREN および CREN ビットをクリアするとデバイスが送信モードになり、それ以外の設定では受信モードになります。RCxSTA レジスタの SPEN ビットをセットすると EUSART が有効になります。

33.5.1.1 マスタクロック

同期データ転送には、データと同期した専用のクロックラインを使います。マスタとして設定されたデバイスが、TX/CK ラインを介してクロックを送信します。TX/CK ピンの出力ドライバは、EUSART を同期送信または受信モードに設定すると自動的に有効になります。シリアルデータ ビットは各クロックの後側のエッジで確実に有効になるように、そのクロックの前側のエッジで遷移します。各データビットに対して1クロックサイクルが生成されます。データビットの数だけクロックサイクルが生成されます。

33.5.1.2 クロック極性

Microwire との互換性を持たせるために、クロック極性を選択できます。クロック極性は、BAUDxCON レジスタの SCKP ビットで選択します。SCKP ビットをセットすると、High がクロックのアイドル状態になります。SCKP ビットをセットすると、各クロックの立ち上がりエッジでデータが変化します。SCKP ビット

をクリアすると、Low がクロックのアイドル状態になります。SCKP ビットをクリアすると、各クロックの立ち上がりエッジでデータが変化します。

33.5.1.3 同期マスタ送信

データはデバイスの RX/DT ピンから送信されます。EUSART を同期マスタ送信モードに設定すると、RX/DT および TX/CK ピン出力ドライバが自動的に有効になります。

TXxREG レジスタに文字を書き込むと送信が開始します。TSR 内に以前の文字が一部でも残っている場合、その文字の最後のビットが送信されるまで新しい文字データは TXxREG に保持されます。これが最初の文字の場合、または TSR 内に以前の文字が全く残っていない場合、TXxREG のデータがただちに TSR レジスタに転送されます。TXxREG から TSR にデータが転送されると、その直後に文字の送信が開始します。各データビットはマスタクロックの前側のエッジで変化し、次のクロックの前側のエッジまで有効です。

Note: TSR レジスタはデータメモリに割り当てられていないため、ユーザからはアクセスできません。

33.5.1.4 同期マスタ送信の実行手順:

1. 必要な baud レートに合わせて、SPxBRGH:SPxBRGL レジスタペアと BRGH および BRG16 の両ビットを初期化する ([セクション 33.4 「EUSART baud レートジェネレータ \(BRG\)」](#) 参照)。
2. SYNC、SPEN、CSRC ビットをセットして同期マスタシリアルポートを有効にする。
3. SREN および CREN ビットをクリアして受信モードを無効にする。
4. TXEN ビットをセットして、送信モードを有効にする。
5. 9ビット送信を行う場合、TX9ビットをセットする。
6. 割り込みが必要な場合、PIE1 レジスタの TXIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
7. 9ビット送信を選択した場合、TX9D ビットに9ビット目を書き込む。
8. TXxREG レジスタにデータを書き込むと送信が開始する。

図 33-10: 同期送信

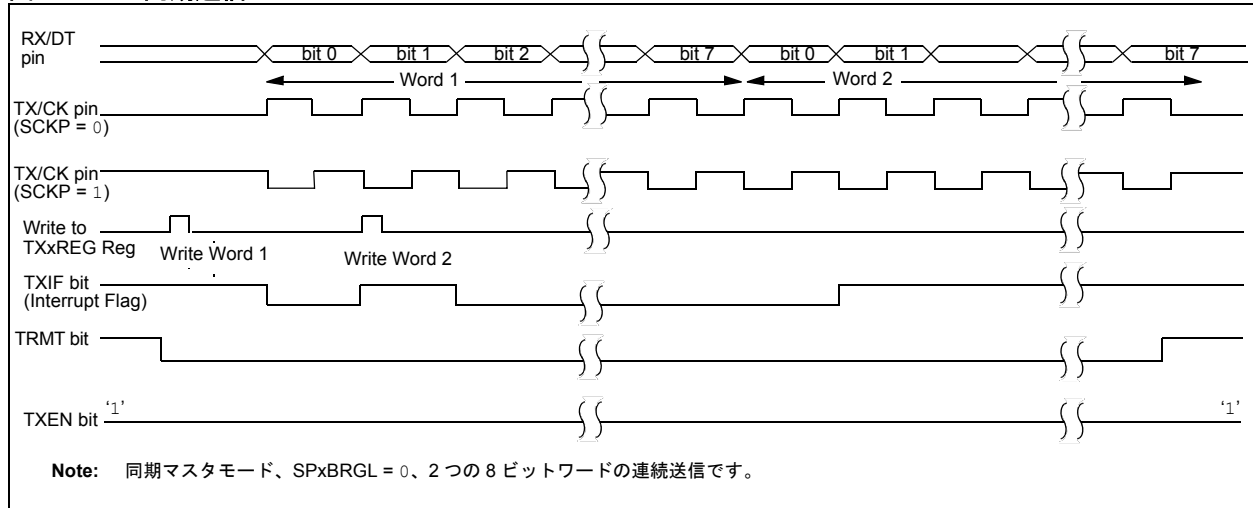
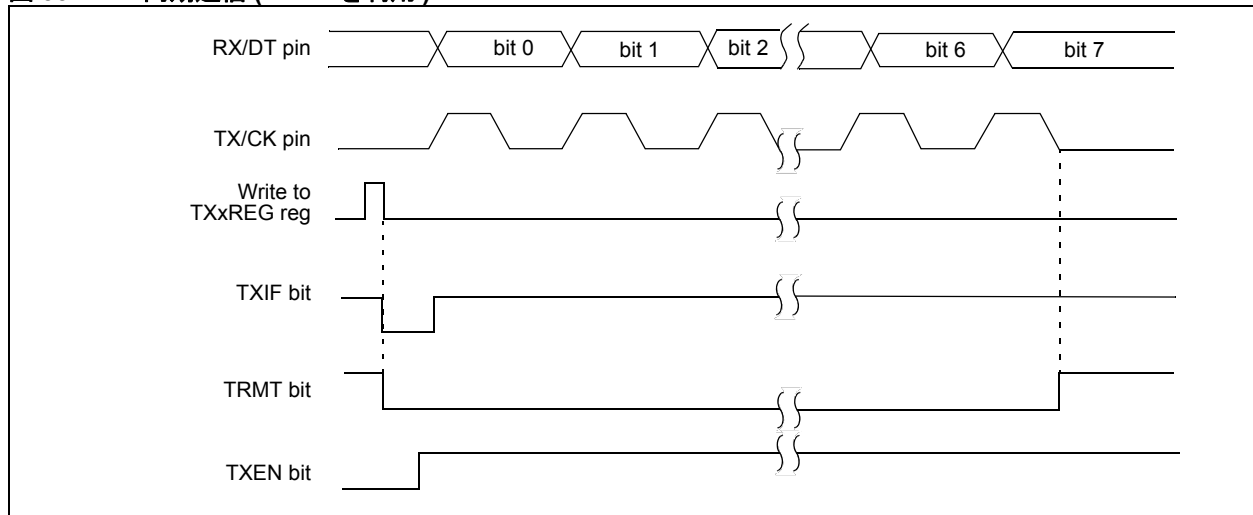


図 33-11: 同期送信 (TXEN を利用)



PIC16(L)F1764/5/8/9

表 33-7: 同期マスタ送信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RxyPPS	—	—	—	RxyPPS<4:0>					158
SP1BRGL	BRG<7:0>								441
SP1BRGH	BRG<15:8>								441
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1REG	EUSART 送信データレジスタ								430*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は同期マスタ送信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9のみです。

2: 未実装、「1」として読み出します。

33.5.1.5 同期マスタ受信

データは RX/DT ピンで受信します。EUSART を同期マスタ受信モードに設定すると、RX/DT ピンの出力ドライバは自動的に無効になります。

同期モードでは、シングル受信イネーブルビット (RCxSTA レジスタの SREN ビット) または連続受信イネーブルビット (RCxSTA レジスタの CREN ビット) のどちらかをセットすると受信が有効になります。

SREN をセットして CREN をクリアすると、1 文字のデータビットの数だけクロックサイクルが生成されます。1 文字の受信が完了すると SREN ビットは自動的にクリアされます。CREN をセットすると、CREN をクリアするまでクロックが連続生成されます。文字の受信中に CREN をクリアすると、CK クロックがただちに停止し、部分的に受信した文字は破棄されます。SREN と CREN を両方セットすると、最初の文字の受信が完了した時点で SREN がクリアされ、以後は CREN の設定が使われます。

受信を開始するには、SREN または CREN のどちらかをセットします。TX/CK クロックピンの後側のエッジで RX/DT ピンのデータをサンプリングし、受信シフトレジスタ (RSR) にシフト入力します。1 文字全体を RSR に受信すると RCIF ビットがセットされ、この文字が 2 文字受信 FIFO に自動的に転送されます。受信 FIFO の先頭文字の下位 8 ビットが RCxREG に格納されます。RCIF ビットは、受信 FIFO 内に読み出していない文字が残っている間はセットされたままです。

Note: RX/DT 機能がアナログピンに割り当てられている場合、対応する ANSEL ビットをクリアしないとレシーバは動作しません。

33.5.1.6 スレーブクロック

同期データ転送には、データと同期した専用のクロックラインを使います。スレーブとして設定されたデバイスは、TX/CK ラインでクロックを受信します。TX/CK ピンの出力ドライバは、デバイスを同期スレーブ送信または受信モードに設定すると自動的に無効になります。シリアルデータ ビットは各クロックの後側のエッジで確実に有効になるように、そのクロックの前側のエッジで遷移します。1 クロックサイクルで 1 ビットのデータを転送します。データビットの数だけクロックサイクルを受信します。

Note: デバイスがスレーブとして設定され、TX/CK 機能がアナログピンに割り当てられている場合、対応する ANSEL ビットをクリアする必要があります。

33.5.1.7 受信オーバーラン エラー

受信 FIFO バッファには 2 文字まで格納できます。RCxREG を読み出して FIFO にアクセスする前に 3 文字目のデータを最後まで受信すると、オーバーランエラーが発生します。オーバーランエラーになると、RCxSTA レジスタの OERR ビットがセットされます。FIFO 内の以前のデータが上書きされる事はありません。既に FIFO バッファ内にある 2 文字は読み出す事ができますが、それ以降の文字はオーバーランエラーが解消されるまで受信できません。OERR ビットは、オーバーラン条件を解消する事によってのみクリアできます。SREN ビットがセット、CREN ビットがクリアの時にオーバーランエラーが発生した場合、RCxREG を読み出すとエラーが解消されます。CREN ビットがセットされている時にオーバーランエラーが発生した場合、RCxSTA レジスタの CREN ビットをクリアするか、SPEN ビットをクリアして EUSART をリセットするとエラー条件が解消されます。

33.5.1.8 9 ビット文字の受信

EUSART は 9 ビット文字の受信をサポートしています。RCxSTA レジスタの RX9 ビットがセットされている場合、EUSART は 1 文字の受信につき 9 ビットを RSR にシフト入力します。RCxSTA レジスタの RX9D ビットには、受信 FIFO の先頭の読み出していない文字の 9 ビット目 (最上位データビット) が格納されます。受信 FIFO バッファから 9 ビットデータを読み出す場合、RCxREG の下位 8 ビットより前に RX9D データビットを読み出す必要があります。

33.5.1.9 同期マスタ受信の実行手順:

1. 適切な baud レートに合わせて、SPxBRGH:SPxBRGL レジスタペアを初期化する。必要に応じて BRGH および BRG16 ビットをセットまたはクリアして、必要な baud レートを設定する。
2. RX ピンの ANSEL ビットをクリアする (必要な場合)。
3. SYNC、SPEN、CSRC ビットをセットして同期マスタ シリアルポートを有効にする。
4. CREN および SREN ビットがクリアされている事を確認する。
5. 割り込みが必要な場合、PIE1 レジスタの RCIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
6. 9 ビット受信を使う場合、RX9 ビットをセットする。
7. SREN ビットをセットするか、連続受信の場合は CREN ビットをセットすると受信が開始する。
8. 1 文字の受信が完了すると、割り込みフラグビット RCIF がセットされる。RCIE 割り込みイネーブルビットをセットしている場合、割り込みが発生する。
9. 9 ビット受信が有効な場合は RCxSTA レジスタを読み出して 9 ビット目を取得し、受信中にエラーが発生したかどうかを確認する。
10. RCxREG レジスタを読み出して、8 ビットの受信データを読み出す。
11. オーバーランエラーが発生した場合、RCxSTA レジスタの CREN ビットをクリアするか、SPEN ビットをクリアして EUSART をリセットしてエラーを解消する。

PIC16(L)F1764/5/8/9

図 33-12: 同期受信 (マスタモード、SREN)

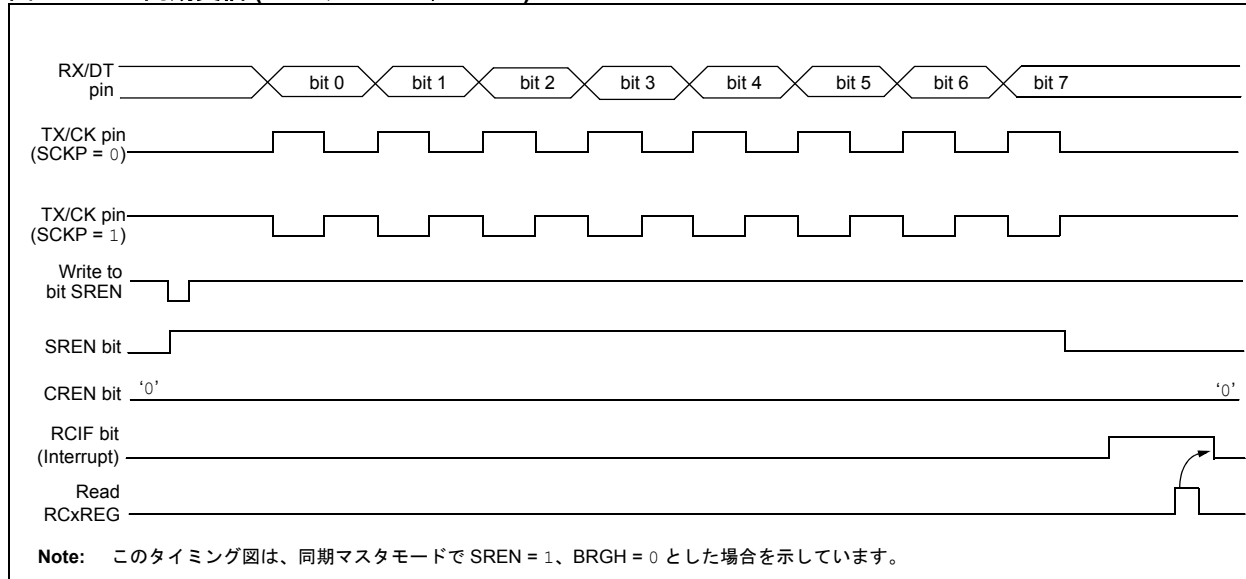


表 33-8: 同期マスタ受信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
CKPPS	—	—	—	CKPPS<4:0>					158、160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1REG	EUSART 受信データレジスタ								433*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RXPPS	—	—	—	RXPPS<4:0>					158、160
RxyPPS	—	—	—	RxyPPS<4:0>					158
SP1BRGL	BRG<7:0>								441*
SP1BRGH	BRG<15:8>								441*
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は同期マスタ受信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9のみです。

2: 未実装、「1」として読み出します。

33.5.2 同期スレーブモード

EUSART モジュールを同期スレーブモードに設定するには、以下のようにビットを設定します。

- SYNC = 1
- CSRC = 0
- SREN = 0 (送信の場合)、SREN = 1 (受信の場合)
- CREN = 0 (送信の場合)、CREN = 1 (受信の場合)
- SPEN = 1

TXxSTA レジスタの SYNC ビットをセットすると、デバイスが同期モードに設定されます。TXxSTA レジスタの CSRC ビットをクリアすると、デバイスがスレーブに設定されます。RCxSTA レジスタの SREN および CREN ビットをクリアするとデバイスが送信モードになり、それ以外の設定では受信モードになります。RCxSTA レジスタの SPEN ビットをセットすると EUSART が有効になります。

33.5.2.1 EUSART 同期スレーブ送信

同期スレーブモードの動作は以下に示すスリープの場合を除いて同期マスターモード ([セクション 33.5.1.3 「同期マスター送信」](#)) と同じです。

同期スレーブモードでは、TXxREG に 2 ワードを書き込んだ後に SLEEP 命令を実行すると、以下の動作が発生します。

1. 最初の 1 文字がただちに TSR レジスタに転送され、送信される。
2. 2 番目のワードは TXxREG レジスタに残る。
3. TXIF ビットはセットされない。
4. 最初の 1 文字が TSR からシフト出力されると、TXxREG レジスタから TSR に 2 番目の文字が転送され、この時点で TXIF ビットがセットされる。
5. PEIE および TXIE ビットが両方共セットされている場合、割り込みによってデバイスがスリープから復帰し、次の命令を実行する。GIE ビットもセットされている場合、プログラムが割り込みサーブスルーチンを呼び出す。

33.5.2.2 同期スレーブ送信の実行手順:

1. SYNC および SPEN ビットをセットし、CSRC ビットをクリアする。
2. CK ピンの ANSEL ビットをクリアする (必要な場合)。
3. CREN および SREN ビットをクリアする。
4. 割り込みが必要な場合、PIE1 レジスタの TXIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
5. 9 ビット送信を行う場合、TX9 ビットをセットする。
6. TXEN ビットをセットして、送信を有効にする。
7. 9 ビット送信を選択した場合、最上位ビット (MSb) を TX9D ビットに書き込む。
8. 下位 8 ビットを TXxREG レジスタに書き込むと送信が開始する。

PIC16(L)F1764/5/8/9

表 33-9: 同期スレーブ送信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
CKPPS	—	—	—	CKPPS<4:0>					158、160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RXPPS	—	—	—	RXPPS<4:0>					158、160
RxyPPS	—	—	—	RxyPPS<4:0>					158
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1REG	EUSART 送信データレジスタ								430*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は同期スレーブ送信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9のみです。

2: 未実装、「1」として読み出します。

33.5.2.3 EUSART 同期スレーブ受信

同期スレーブモードは同期マスタモードと以下の項目で動作が異なりますが、その他は同じです。([セクション 33.5.1.5「同期マスタ受信」](#) 参照)。

- スリープ
- CREN ビット (スレーブモードでは常にセットされるため、レシーバが停止しない)
- SREN ビット (スレーブモードでは「ドントケア」)

スリープに移行する前にCRENビットをセットしておく、スリープ中でも文字を受信できます。文字を受信すると、RSRレジスタからRCxREGレジスタへデータが転送されます。RCIE イネーブルビットがセットされている場合、割り込みによってデバイスがスリープから復帰し、次の命令を実行します。GIE ビットもセットされている場合、プログラムが割り込みベクタに分岐します。

33.5.2.4 同期スレーブ受信の実行手順:

1. SYNC および SPEN ビットをセットし、CSRC ビットをクリアする。
2. CK および DT ピン両方の ANSEL ビットをクリアする (必要な場合)。
3. 割り込みが必要な場合、PIE1 レジスタの RCIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
4. 9ビット受信を使う場合、RX9ビットをセットする。
5. CREN ビットをセットして受信を有効にする。
6. 受信が完了すると、RCIF ビットがセットされる。RCIE ビットをセットしている場合、割り込みが発生する。
7. 9 ビットモードを有効にしている場合、RCxSTA レジスタの RX9D ビットから最上位ビット (MSb) を取得する。
8. RCxREG レジスタを読み出し、受信 FIFO から下位 8 ビットを取得する。
9. オーバーラン エラーが発生した場合、RCxSTA レジスタの CREN ビットをクリアするか、SPEN ビットをクリアして EUSART をリセットしてエラーを解消する。

表 33-10: 同期スレーブ受信関連レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	レジスタ 内容記載 ページ
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
CKPPS	—	—	—	CKPPS<4:0>					158、160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1REG	EUSART 受信データレジスタ								433*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RXPPS	—	—	—	RXPPS<4:0>					158、160
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

凡例: — = 未実装、「0」として読み出し。網掛けの部分は同期スレーブ受信では使いません。

* 本文中でレジスタを説明しているページです。

Note 1: PIC16(L)F1768/9 のみです。

2: 未実装、「1」として読み出します。

33.6 スリープ中の EUSART の動作

EUSART は、同期スリープモードの場合のみスリープ中でも動作を継続します。その他のモードではシステムクロックが必要なため、スリープ中は送受信シフトレジスタの動作に必要な信号を生成できません。

同期スリープモードでは、外部で生成されたクロックを使って送受信シフトレジスタを動作させる事ができます。

33.6.1 スリープ中の同期受信

スリープ中に受信するには、スリープに移行する前に以下の条件を全て満たす必要があります。

- 制御レジスタの RCxSTA と TXxSTA を同期スリープ受信モードに設定する ([セクション 33.5.2.4 「同期スリープ受信の実行手順」](#) 参照)。
- 割り込みが必要な場合、PIE1 レジスタの RCIE ビットと INTCON レジスタの GIE および PEIE ビットをセットする。
- RCxREG を読み出して受信バッファ内の保留中の文字を全て空にし、RCIF 割り込みフラグをクリアする。

この状態でスリープに移行すると、デバイスは RX/DT および TX/CK ピンでそれぞれデータとクロックを受信できる状態になります。外部デバイスが生成するクロックによってデータワードが完全に入力されると、PIR1 レジスタの RCIF 割り込みフラグビットがセットされます。これによって、プロセッサがスリープから復帰します。

スリープから復帰すると、SLEEP 命令の次の命令が実行されます。INTCON レジスタの GIE (グローバル割り込みイネーブル) ビットもセットされている場合、割り込みサービスルーチン (アドレス 004h) が呼び出されます。

33.6.2 スリープ中の同期送信

スリープ中に送信するには、スリープに移行する前に以下の条件を全て満たす必要があります。

- 制御レジスタの RCxSTA と TXxSTA を同期スリープ送信モードに設定する ([セクション 33.5.2.2 「同期スリープ送信の実行手順」](#) 参照)。
- TXxREG へ出力データを書き込み、TSR と送信バッファにデータを格納して TXIF 割り込みフラグをクリアする。
- 割り込みが必要な場合、PIE1 レジスタの TXIE ビットと INTCON レジスタの PEIE ビットをセットする。
- 割り込みイネーブルビットである PIE1 レジスタの TXIE ビットと INTCON レジスタの PEIE ビットをセットする。

この状態でスリープに移行すると、デバイスは TX/CK ピンでクロックを受信でき、RX/DT ピンでデータを送信できる状態になります。外部デバイスのクロックによって TSR に格納されていたデータワードが全て出力されると、TXxREG に保留されていたバイトが TSR へ転送されて TXIF フラグがセットされます。これによって、プロセッサがスリープから復帰します。この時点で、TXxREG は次に送信する文字データを格納できる状態になります。TXxREG にデータを書き込むと TXIF フラグがクリアされます。

スリープから復帰すると、SLEEP 命令の次の命令が実行されます。GIE (グローバル割り込みイネーブル) ビットもセットされている場合、割り込みサービスルーチン (アドレス 0004h) が呼び出されます。

34.0 インサーキット シリアルプログラミング™(ICSP™)

ICSP™ プログラミングを使うと、未書き込みのデバイスを使って回路基板を生産できます。組み立てプロセス後に書き込む事で、最新ファームウェアまたはカスタムファームウェアを使ってデバイスをプログラムできます。ICSP™ プログラミングには、以下の5本のピンが必要です。

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}/\text{VPP}$
- VDD
- VSS

プログラム / ベリファイモードを使って、プログラムメモリ、ユーザID、コンフィグレーションワードをシリアル通信を介してプログラムします。ICSPDATピンはシリアルデータ転送用の双方向 I/O であり、ICSPCLK ピンはクロック入力です。ICSP™ の詳細は、『PIC16(L)F170X Memory Programming Specification』(DS40001683) を参照してください。

34.1 高電圧プログラミング エントリモード

ICSPCLK ピンと ICSPDAT ピンを Low に保持し、 $\overline{\text{MCLR}}/\text{VPP}$ の電圧を V_{IH} にすると、デバイスは高電圧プログラミング エントリモードに移行します。

34.2 低電圧プログラミング エントリモード

低電圧プログラミング エントリモードでは、高電圧を使わずに VDD だけを使って PIC® フラッシュ MCU をプログラムできます。コンフィグレーションワードの LVP ビットを「1」にセットすると、低電圧 ICSP プログラミング エントリが有効になります。低電圧 ICSP モードを無効にするには、LVP ビットを「0」にクリアします。

低電圧プログラミング エントリモードに移行するステップは以下の通りです。

1. $\overline{\text{MCLR}}$ を V_{IL} にする。
2. 32ビットのキーシーケンスを、ICSPCLK をクロッキングしながら ICSPDAT に入力する。

キーシーケンスの完了後、プログラム / ベリファイモードを継続する期間、 $\overline{\text{MCLR}}$ を V_{IL} に保持する必要があります。

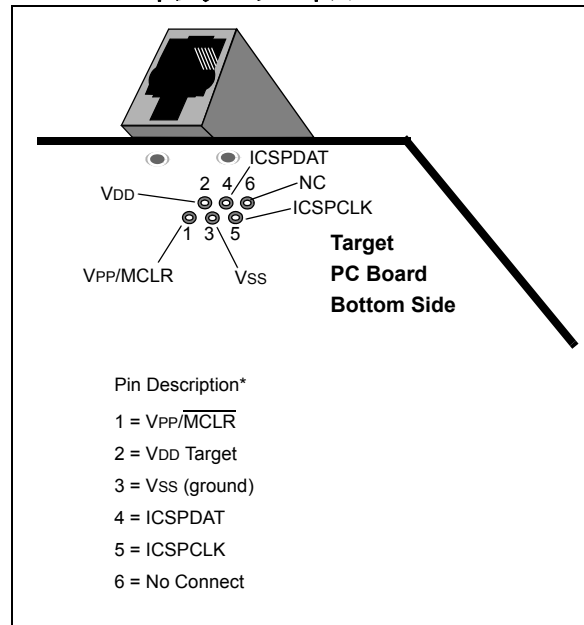
低電圧プログラミングの有効時 (LVP = 1) は、 $\overline{\text{MCLR}}$ リセット機能が自動的に有効になり、無効にできません。詳細は**セクション 6.5 「 $\overline{\text{MCLR}}$ 」**を参照してください。

再度 LVP ビットを「0」にクリアするには、高電圧プログラミングモードを使う必要があります。

34.3 一般的なプログラミングインターフェイス

ターゲットデバイスへの接続は、通常は ICSP™ ヘッダを介して行います。開発ツールによく使われるコネクタは、6P6C (6 ピン、6 コネクタ) 構成の RJ-11 です。**図 34-1** を参照してください。

図 34-1: ICD RJ-11 型コネクタインターフェイス



また、PICKit™ プログラマによく使われるコネクタとして、0.1 インチ間隔の標準 6 ピンヘッダがあります。**図 34-2** を参照してください。

インターフェイスに関するその他の推奨事項は、プリント基板の設計前に、ご使用のデバイスプログラマのマニュアルを参照してください。

絶縁デバイスを使って、プログラミングピンと他の回路を絶縁する事を推奨します。絶縁方法はアプリケーション固有であり、抵抗、ダイオード等のデバイスや、ジャンパを使う事もあります。詳細は**図 34-3** を参照してください。

PIC16(L)F1764/5/8/9

図 34-2: PICKit™ プログラマ型コネクタ インターフェイス

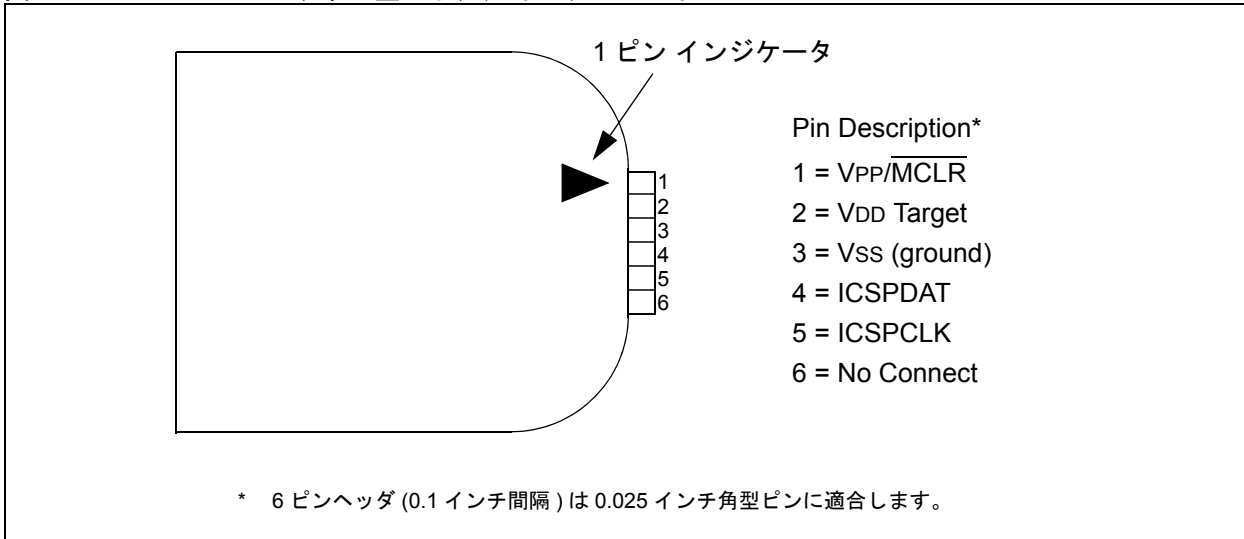
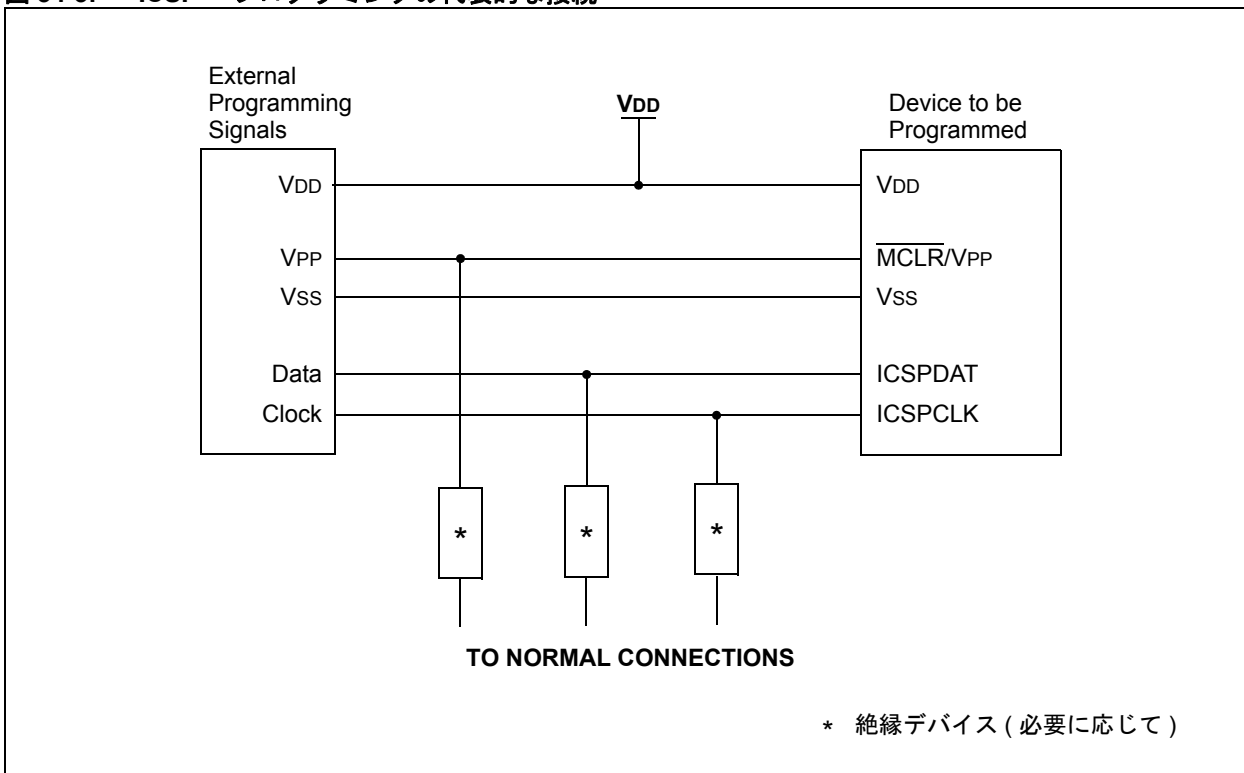


図 34-3: ICSP™ プログラミングの代表的な接続



35.0 命令セットのまとめ

各命令は 14 ビットワードで構成されており、オペコードとそれらに必要なオペランドを全て含みます。オペコードは、主に 3 つのカテゴリに分けられます。

- バイト指向命令
- ビット指向命令
- リテラルおよび制御命令

リテラルと制御のカテゴリには、最も多くの種類の命令ワード形式があります。

表 35-3 に、MPASM™ アセンブラが認識する命令の一覧を示します。

全ての命令は、1 命令サイクルで実行されます。しかし、以下に示す例外は 2 サイクルまたは 3 サイクルを必要とします。

- サブルーチンは 2 サイクルを必要とする (CALL、CALLW)。
- 割り込みまたはサブルーチンからの戻りは 2 サイクルを必要とする (RETURN、RETLW、RETFIE)。
- プログラムの分岐は 2 サイクルを必要とする (GOTO、BRA、BRW、BTFSS、BTFSC、DECFSZ、INCSFZ)。
- 命令が間接ファイルレジスタを参照し、ファイルセレクトレジスタがプログラムメモリを指している場合、追加の 1 命令サイクルを必要とする。

1 命令サイクルはオシレータ 4 周期分です。例えばオシレータ周波数が 4 MHz の場合の命令実行周波数の公称値は 1 MHz です。

命令の例では、「0xhh」の形式で 16 進数を表しています。この「h」は 16 進数の 1 桁を表します。

35.1 Read-Modify-Write 動作

命令の一部としてファイルレジスタを指定するような命令は、必ず Read-Modify-Write (R-M-W) 動作を実行します。レジスタを読み出し、データを変更し、そして命令または格納先指定文字「d」のどちらかに従って結果を書き込みます。書き込み先がそのレジスタであっても読み出し動作を実行します。

表 35-1: オペコード フィールドの説明

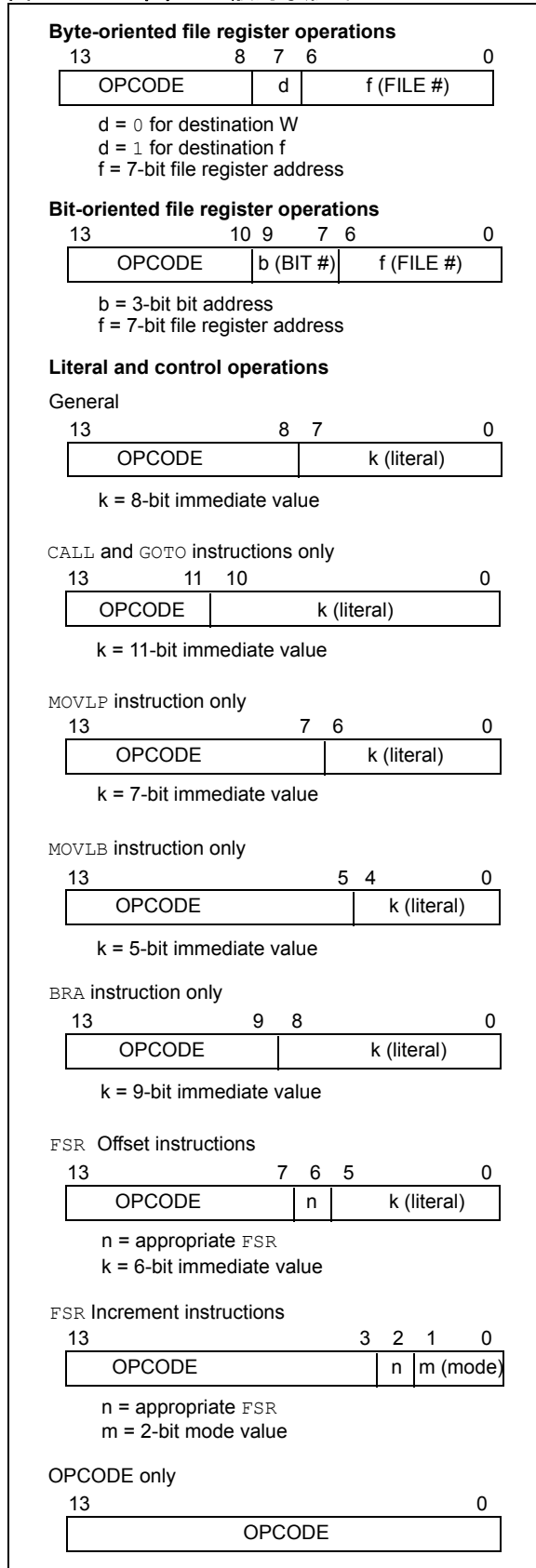
フィールド	説明
f	レジスタ ファイルアドレス (0x00 ~ 0x7F)
W	ワーキングレジスタ (アキュムレータ)
b	8 ビット ファイルレジスタのビットアドレス
k	リテラル フィールド、定数データ、ラベル
x	ドントケア ロケーション (= 0 または 1)、アセンブラは x = 0 でコードを生成します (Microchip社の全ソフトウェア ツールとの互換性を確保するためにこの形式を推奨します)。
d	格納先の選択。d = 0: 結果を W に格納します。 d = 1: 結果をファイルレジスタ f に格納します。 既定値は d = 1 です。
n	FSR または INDF 番号 (0、1)
mm	(プリ/ポスト) インクリメント / デクリメント モードの選択

表 35-2: 略語の説明

フィールド	説明
PC	プログラム カウンタ
\overline{TO}	タイムアウト ビット
C	キャリービット
DC	ディジット キャリービット
Z	ゼロビット
\overline{PD}	パワーダウン ビット

PIC16(L)F1764/5/8/9

図 35-1: 命令の一般的な形式



PIC16(L)F1764/5/8/9

表 35-3: PIC16(L)F1764/5/8/9 の命令セット

ニーモニック、オペランド	説明	サイクル数	14ビットオペコード				影響を受けるステータス	Note	
			MSb	LSb					
バイト指向ファイルレジスタ命令									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	—	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	Z	2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	Z	2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
バイト指向スキップ命令									
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00	1111	dfff	ffff		1, 2
ビット指向ファイルレジスタ命令									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
ビット指向スキップ命令									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
リテラル命令									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLW	k	Move literal to W	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note** 1: プログラム カウンタ (PC) が変更された場合、あるいは条件付きテストの結果が真の場合、命令実行には2サイクルが必要です。2サイクル目は、NOPとして実行されます。
- 2: この命令が INDF レジスタをアドレス指定し、対応する FSR の MSb がセットされている場合、追加の1命令サイクルが必要です。

PIC16(L)F1764/5/8/9

表 35-3: PIC16(L)F1764/5/8/9 の命令セット (続き)

ニーモニック、 オペランド	説明	サイクル 数	14 ビットオペコード			影響を 受ける ステータス	Note		
			MSb	LSb					
制御命令									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	–	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
固有命令									
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	\overline{TO} , \overline{PD}	
NOP	–	No Operation	1	00	0000	0000	0000		
OPTION	–	Load OPTION_REG register with W	1	00	0000	0110	0010		
RESET	–	Software device Reset	1	00	0000	0000	0001		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	\overline{TO} , \overline{PD}	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C コンパイラに最適化された命令									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm	Z	2, 3
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	kkkk	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00	0000	0001	1nmm		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk	kkkk		2

- Note** 1: プログラム カウンタ (PC) が変更された場合、あるいは条件付きテストの結果が真の場合、命令実行には 2 サイクルが必要です。2 サイクル目は、NOP として実行されます。
- 2: この命令が INDF レジスタをアドレス指定し、対応する FSR の MSb がセットされている場合、追加の 1 命令サイクルが必要です。
- 3: MOVIW 命令と MOVWI 命令の説明の表を参照してください。

35.2 命令の説明

ADDFSR Add Literal to FSRn

構文: `[label] ADDFSR FSRn, k`
 オペランド: $-32 \leq k \leq 31$
 $n \in [0, 1]$
 動作: $FSR(n) + k \rightarrow FSR(n)$
 影響を受けるステータス: なし
 説明: 符号付き 6 ビットリテラル「k」を FSRnH:FSRnL レジスタペアの内容に加算します。

FSRn の範囲は 0000h ~ FFFFh です。これらの範囲を超えると、FSR のラップアラウンド (折り返し) が発生します。

ADDLW Add literal and W

構文: `[label] ADDLW k`
 オペランド: $0 \leq k \leq 255$
 動作: $(W) + k \rightarrow (W)$
 影響を受けるステータス: C, DC, Z
 説明: W レジスタの内容を 8 ビットのリテラル「k」に加算し、結果を W レジスタに書き込みます。

ADDWF Add W and f

構文: `[label] ADDWF f, d`
 オペランド: $0 \leq f \leq 127$
 $d \in [0, 1]$
 動作: $(W) + (f) \rightarrow (\text{destination})$
 影響を受けるステータス: C, DC, Z
 説明: W レジスタとレジスタ「f」の内容を加算します。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

ADDWFC ADD W and CARRY bit to f

構文: `[label] ADDWFC f{,d}`
 オペランド: $0 \leq f \leq 127$
 $d \in [0, 1]$
 動作: $(W) + (f) + (C) \rightarrow \text{dest}$
 影響を受けるステータス: C, DC, Z
 説明: W レジスタ、キャリーフラグ、データメモリのアドレス「f」の内容を加算します。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果をデータメモリのアドレス「f」に格納します。

ANDLW AND literal with W

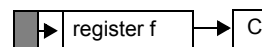
構文: `[label] ANDLW k`
 オペランド: $0 \leq k \leq 255$
 動作: $(W) .AND.(k) \rightarrow (W)$
 影響を受けるステータス: Z
 説明: W レジスタの内容と 8 ビットのリテラル「k」を AND 演算します。結果は W レジスタに書き込まれます。

ANDWF AND W with f

構文: `[label] ANDWF f, d`
 オペランド: $0 \leq f \leq 127$
 $d \in [0, 1]$
 動作: $(W) .AND.(f) \rightarrow (\text{destination})$
 影響を受けるステータス: Z
 説明: W レジスタとレジスタ「f」を AND 演算します。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

ASRF Arithmetic Right Shift

構文: `[label] ASRF f{,d}`
 オペランド: $0 \leq f \leq 127$
 $d \in [0, 1]$
 動作: $(f < 7) \rightarrow \text{dest} < 7$ 、
 $(f < 7:1) \rightarrow \text{dest} < 6:0$ 、
 $(f0) \rightarrow C$
 影響を受けるステータス: C, Z
 説明: レジスタ「f」の内容を、キャリーフラグを通して右へ 1 ビット移動させます。MSb は変更しません。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。



PIC16(L)F1764/5/8/9

BCF Bit Clear f

構文: [label] BCF f,b
オペランド: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
動作: $0 \rightarrow (f < b >)$
影響を受けるステータス: なし
説明: レジスタ「f」のビット「b」をクリアします。

BTFSC Bit Test f, Skip if Clear

構文: [label] BTFSC f,b
オペランド: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
動作: skip if $(f < b >) = 0$
影響を受けるステータス: なし
説明: レジスタ「f」のビット「b」が「1」の場合、次の命令を実行します。レジスタ「f」のビット「b」が「0」の場合、次の命令を破棄し、代わりにNOPを実行して、2サイクルの命令にします。

BRA Relative Branch

構文: [label] BRA label
[label] BRA \$+k
オペランド: $-256 \leq \text{label} - \text{PC} + 1 \leq 255$
 $-256 \leq k \leq 255$
動作: $(\text{PC}) + 1 + k \rightarrow \text{PC}$
影響を受けるステータス: なし
説明: 符号付き9ビットリテラル「k」をPCに加算します。PCはインクリメントして次の命令をフェッチするため、新しいアドレスは $\text{PC} + 1 + (k)$ です。この命令は2サイクル命令です。この分岐には範囲の制限があります。

BTFSS Bit Test f, Skip if Set

構文: [label] BTFSS f,b
オペランド: $0 \leq f \leq 127$
 $0 \leq b < 7$
動作: skip if $(f < b >) = 1$
影響を受けるステータス: なし
説明: レジスタ「f」のビット「b」が「0」の場合、次の命令を実行します。ビット「b」が「1」の場合、次の命令を破棄し、代わりにNOPを実行して、2サイクルの命令にします。

BRW Relative Branch with W

構文: [label] BRW
オペランド: なし
動作: $(\text{PC}) + (W) \rightarrow \text{PC}$
影響を受けるステータス: なし
説明: W(符号なし)の内容をPCに加算します。PCはインクリメントして次の命令をフェッチするため、新しいアドレスは $\text{PC} + 1 + (W)$ です。この命令は2サイクル命令です。

BSF Bit Set f

構文: [label] BSF f,b
オペランド: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
動作: $1 \rightarrow (f < b >)$
影響を受けるステータス: なし
説明: レジスタ「f」のビット「b」をセットします。

CALL Call Subroutine

構文: [label] CALL k
 オペランド: $0 \leq k \leq 2047$
 動作: (PC) + 1 → TOS,
 k → PC<10:0>、
 (PCLATH<6:3>) → PC<14:11>
 影響を受けるステータス: なし
 説明: サブルーチン呼び出し。最初に、リターンアドレス (PC + 1) をスタックにプッシュします。11 ビットの即値アドレスを、PC ビット <10:0> に書き込みます。PC の上位ビットは、PCLATH から書き込まれます。CALL は 2 サイクルの命令です。

CALLW Subroutine Call With W

構文: [label] CALLW
 オペランド: なし
 動作: (PC) + 1 → TOS、
 (W) → PC<7:0>、
 (PCLATH<6:0>) → PC<14:8>
 影響を受けるステータス: なし
 説明: W レジスタを使うサブルーチン呼び出しです。まず、リターンアドレス (PC+1) をリターンスタックへプッシュします。次に W レジスタの内容を PC<7:0> に書き込み、PCLATH の内容を PC<14:8> に書き込みます。CALLW は 2 サイクルの命令です。

CLRF Clear f

構文: [label] CLRF f
 オペランド: $0 \leq f \leq 127$
 動作: 00h → (f)、
 1 → Z
 影響を受けるステータス: Z
 説明: レジスタ「f」の内容をクリアして、Z ビットをセットします。

CLRW Clear W

構文: [label] CLRW
 オペランド: なし
 動作: 00h → (W)、
 1 → Z
 影響を受けるステータス: Z
 説明: W レジスタをクリアします。ゼロビット (Z) をセットします。

CLRWDWDT Clear Watchdog Timer

構文: [label] CLRWDWDT
 オペランド: なし
 動作: 00h → WDT、
 0 → WDT prescaler、
 1 → \overline{TO} 、
 1 → PD
 影響を受けるステータス: \overline{TO} 、 \overline{PD}
 説明: CLRWDWDT 命令はウォッチドッグ タイマをリセットします。WDT のプリスケールもリセットします。ステータスビット \overline{TO} と PD をセットします。

COMF Complement f

構文: [label] COMF f,d
 オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
 動作: $\overline{(f)}$ → (destination)
 影響を受けるステータス: Z
 説明: レジスタ「f」の内容の補数をとります。「d」が「0」の場合は結果が W レジスタに書き込まれ、「d」が「1」の場合は結果がレジスタ「f」に書き戻されます。

DECF Decrement f

構文: [label] DECF f,d
 オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
 動作: (f) - 1 → (destination)
 影響を受けるステータス: Z
 説明: レジスタ「f」の内容をデクリメントします。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

PIC16(L)F1764/5/8/9

DECFSZ Decrement f, Skip if 0

構文: [label] DECFSZ f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: $(f) - 1 \rightarrow (\text{destination})$,
skip if result = 0
影響を受けるステータス: なし
説明: レジスタ「f」の内容をデクリメントします。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。結果が「1」の場合、次の命令が実行されます。結果が「0」の場合、代わりに NOP を実行して、2 サイクルの命令にします。

GOTO Unconditional Branch

構文: [label] GOTO k
オペランド: $0 \leq k \leq 2047$
動作: $k \rightarrow \text{PC} \langle 10:0 \rangle$,
 $\text{PCLATH} \langle 6:3 \rangle \rightarrow \text{PC} \langle 14:11 \rangle$
影響を受けるステータス: なし
説明: GOTO は無条件分岐です。11 ビットの即値を、PC ビット $\langle 10:0 \rangle$ に書き込みます。PC の上位ビットは、PCLATH $\langle 4:3 \rangle$ から書き込まれます。GOTO は 2 サイクルの命令です。

INCF Increment f

構文: [label] INCF f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: $(f) + 1 \rightarrow (\text{destination})$
影響を受けるステータス: Z
説明: レジスタ「f」の内容をインクリメントします。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

INCFSZ Increment f, Skip if 0

構文: [label] INCFSZ f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: $(f) + 1 \rightarrow (\text{destination})$,
skip if result = 0
影響を受けるステータス: なし
説明: レジスタ「f」の内容をインクリメントします。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。結果が「1」の場合、次の命令が実行されます。結果が「0」の場合には代わりに NOP を実行して、2 サイクルの命令にします。

IORLW Inclusive OR literal with W

構文: [label] IORLW k
オペランド: $0 \leq k \leq 255$
動作: $(W) .OR. k \rightarrow (W)$
影響を受けるステータス: Z
説明: W レジスタの内容と 8 ビットのリテラル「k」を OR 演算します。結果は W レジスタに書き込まれます。

IORWF Inclusive OR W with f

構文: [label] IORWF f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: $(W) .OR. (f) \rightarrow (\text{destination})$
影響を受けるステータス: Z
説明: W レジスタとレジスタ「f」を OR 演算します。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

LSLF Logical Left Shift

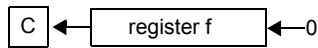
構文: `[label] LSLF f{,d}`

オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$

動作: $(f<7>) \rightarrow C$,
 $(f<6:0>) \rightarrow \text{dest}<7:1>$,
 $0 \rightarrow \text{dest}<0>$

影響を受ける
ステータス: C, Z

説明: レジスタ「f」の内容を、キャリーフラグを通して左へ1ビット移動させます。LSbに「0」が入ります。「d」が「0」の場合、結果はWレジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。



LSRF Logical Right Shift

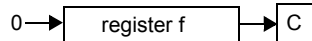
構文: `[label] LSRF f{,d}`

オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$

動作: $0 \rightarrow \text{dest}<7>$,
 $(f<7:1>) \rightarrow \text{dest}<6:0>$,
 $(f0) \rightarrow C$

影響を受ける
ステータス: C, Z

説明: レジスタ「f」の内容を、キャリーフラグを通して右へ1ビット移動させます。MSbに「0」が入ります。「d」が「0」の場合、結果はWレジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。



MOVF Move f

構文: `[label] MOVF f,d`

オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$

動作: $(f) \rightarrow (\text{destination})$

影響を受ける
ステータス: Z

説明: レジスタ「f」の内容を、「d」で示された格納先に移動します。「d」=「0」の場合、格納先はWレジスタです。「d」=「1」の場合、格納先はファイルレジスタ「f」自身です。ステータスフラグZは影響を受けるため、ファイルレジスタのテストには「d」=「1」を使います。

ワード数: 1

サイクル数: 1

例: `MOVF FSR, 0`

命令実行後

W = FSRレジスタの値

Z = 1

PIC16(L)F1764/5/8/9

MOVIW Move INDFn to W

構文: [label] MOVIW ++FSRn
[label] MOVIW -FSRn
[label] MOVIW FSRn++
[label] MOVIW FSRn--
[label] MOVIW k[FSRn]

オペランド: $n \in [0,1]$
 $mm \in [00,01,10,11]$
 $-32 \leq k \leq 31$

動作: INDFn → W
有効なアドレスは、下式で決定されます。

- FSR + 1(プリ インクリメント)
- FSR - 1(プリ デクリメント)
- FSR + k(相対オフセット)

移動後、FSR は以下の値のどれかを取ります。

- FSR + 1(全てインクリメント)
- FSR - 1(全てデクリメント)
- 不変

影響を受けるステータス: Z

モード	構文	mm
プリ インクリメント	++FSRn	00
プリ デクリメント	-FSRn	01
ポスト インクリメント	FSRn++	10
ポスト デクリメント	FSRn--	11

説明: この命令は、W レジスタから間接レジスタ (INDFn) ヘデータを移動する際に使います。この動作の前後では、プリ (ポスト) インクリメント / デクリメントでポインタ (FSRn) がアップデートされます。

Note: INDFn レジスタは物理的なレジスタではありません。INDFn レジスタにアクセスする命令は、実際には FSRn で指定したアドレスにあるレジスタ値にアクセスしています。

FSRn のレンジは 0000h ~ FFFFh です。インクリメント / デクリメントでこの範囲を超えると、ラップアラウンドが発生します。

MOVLB Move literal to BSR

構文: [label] MOVLB k

オペランド: $0 \leq k \leq 31$

動作: $k \rightarrow \text{BSR}$

影響を受けるステータス: なし

説明: 5 ビットのリテラル「k」をバンクセレクト レジスタ (BSR) に書き込みます。

MOVLP Move literal to PCLATH

構文: [label] MOVLP k

オペランド: $0 \leq k \leq 127$

動作: $k \rightarrow \text{PCLATH}$

影響を受けるステータス: なし

説明: 7 ビットのリテラル「k」を PCLATH レジスタに書き込みます。

MOVLW Move literal to W

構文: [label] MOVLW k

オペランド: $0 \leq k \leq 255$

動作: $k \rightarrow (W)$

影響を受けるステータス: なし

説明: 8 ビットのリテラル「k」を W レジスタに書き込みます。「ドントケア」は「0」としてアセンブルされます。

ワード数: 1

サイクル数: 1

例: `MOVLW 0x5A`
命令実行後
`W = 0x5A`

MOVWF Move W to f

構文: [label] MOVWF f

オペランド: $0 \leq f \leq 127$

動作: $(W) \rightarrow (f)$

影響を受けるステータス: なし

説明: W レジスタから、レジスタ「f」にデータを移動します。

ワード数: 1

サイクル数: 1

例: `MOVWF OPTION_REG`
命令実行前
`OPTION_REG = 0xFF`
`W = 0x4F`
命令実行後
`OPTION_REG = 0x4F`
`W = 0x4F`

MOVWI	Move W to INDFn
構文:	[label] MOVWI ++FSRn [label] MOVWI --FSRn [label] MOVWI FSRn++ [label] MOVWI FSRn-- [label] MOVWI k[FSRn]
オペランド:	$n \in [0,1]$ $mm \in [00,01,10,11]$ $-32 \leq k \leq 31$
動作:	$W \rightarrow \text{INDFn}$ 有効なアドレスは、下式で決定されます。 <ul style="list-style-type: none"> FSR + 1(プリ インクリメント) FSR - 1(プリ デクリメント) FSR + k(相対オフセット) 移動後、FSR は以下の値のどれかを取り ます。 <ul style="list-style-type: none"> FSR + 1(全てインクリメント) FSR - 1(全てデクリメント) 不変
影響を受けるステータス:	なし

モード	構文	mm
プリ インクリメント	++FSRn	00
プリ デクリメント	--FSRn	01
ポストインクリメント	FSRn++	10
ポストデクリメント	FSRn--	11

説明: この命令は、W レジスタから間接レジスタ (INDFn) ヘデータを移動する際に使います。この動作の前後では、プリ (ポスト) インクリメント/デクリメントでポインタ (FSRn) がアップデートされます。

Note: INDFn レジスタは物理的なレジスタではありません。INDFn レジスタにアクセスする命令は、実際には FSRn で指定したアドレスにあるレジスタ値にアクセスしています。

FSRn の範囲は 0000h ~ FFFFh です。インクリメント/デクリメントでこの範囲を超えると、ラップアラウンドが発生します。

FSRn でのインクリメント/デクリメントは、どのステータスビットにも影響しません。

NOP	No Operation
構文:	[label] NOP
オペランド:	なし
動作:	なし
影響を受けるステータス:	なし
説明:	何も実行しません。
ワード数:	1
サイクル数:	1
例:	NOP

OPTION	Load OPTION_REG Register with W
構文:	[label] OPTION
オペランド:	なし
動作:	(W) → OPTION_REG
影響を受けるステータス:	なし
説明:	W レジスタから、OPTION_REG レジスタにデータを移動します。
ワード数:	1
サイクル数:	1
例:	OPTION 命令実行前 OPTION_REG = 0xFF W = 0x4F 命令実行後 OPTION_REG = 0x4F W = 0x4F

RESET	Software Reset
構文:	[label] RESET
オペランド:	なし
動作:	デバイスをリセットします。PCON レジスタの $\overline{\text{RI}}$ フラグをリセットします。
影響を受けるステータス:	なし
説明:	この命令は、ソフトウェアでハードウェアリセットを実行する方法を提供します。

PIC16(L)F1764/5/8/9

RETFIE Return from Interrupt

構文: [label] RETFIE k

オペランド: なし

動作: TOS → PC,
1 → GIE

影響を受けるステータス: なし

説明: 割り込み処理から復帰します。スタックがポップされ、Top-of-Stack (TOS) が PC に書き込まれます。グローバル割り込みイネーブルビット GIE (INTCON<7>) をセットして、割り込みを有効にします。これは 2 サイクルの命令です。

ワード数: 1

サイクル数: 2

例: RETFIE

割り込み後
PC = TOS
GIE = 1

RETLW Return with literal in W

構文: [label] RETLW k

オペランド: $0 \leq k \leq 255$

動作: k → (W),
TOS → PC

影響を受けるステータス: なし

説明: 8 ビットのリテラル「k」を W レジスタに書き込みます。Top-of-Stack(リターンアドレス) をプログラム カウンタに書き込みます。これは 2 サイクルの命令です。

ワード数: 1

サイクル数: 2

例: CALL TABLE; W contains table
; offset value
; W now has table value
TABLE
.
.
ADDWF PC ; W = offset
RETLW k1 ; Begin table
RETLW k2 ;
.
.
RETLW kn ; End of table

命令実行前
W = 0x07

命令実行後
W = k8 の値

RETURN Return from Subroutine

構文: [label] RETURN

オペランド: なし

動作: TOS → PC

影響を受けるステータス: なし

説明: サブルーチンから戻ります。スタックがポップされ、Top-of-Stack (TOS) がプログラム カウンタに書き込まれます。これは 2 サイクルの命令です。

RLF Rotate Left f through Carry

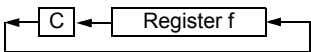
構文: [label] RLF f,d

オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$

動作: 下記参照

影響を受けるステータス: C

説明: レジスタ「f」の内容を、キャリーフラグを通して左回りに 1 ビット移動させます。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。



ワード数: 1

サイクル数: 1

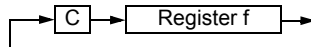
例: RLF REG1,0

命令実行前
REG1 = 1110 0110
C = 0

命令実行後
REG1 = 1110 0110
W = 1100 1100
C = 1

RRF Rotate Right f through Carry

構文: `[label] RRF f,d`
 オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
 動作: 下記参照
 影響を受けるステータス: C
 説明: レジスタ「f」の内容を、キャリーフラグを通して右回りに1ビット移動させます。「d」が「0」の場合、結果はWレジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。



SLEEP Enter Sleep mode

構文: `[label] SLEEP`
 オペランド: なし
 動作: $00h \rightarrow$ WDT,
 $0 \rightarrow$ WDT prescaler,
 $1 \rightarrow \overline{TO}$,
 $0 \rightarrow \overline{PD}$
 影響を受けるステータス: \overline{TO} , \overline{PD}
 説明: パワーダウンステータスビット (\overline{PD}) をクリアします。タイムアウトステータスビット (\overline{TO}) をセットします。ウォッチドッグタイマとそのプリスケアラをクリアします。オシレータを停止してプロセッサをスリープにします。

SUBLW Subtract W from literal

構文: `[label] SUBLW k`
 オペランド: $0 \leq k \leq 255$
 動作: $k - (W) \rightarrow (W)$
 影響を受けるステータス: C, DC, Z
 説明: 8ビットのリテラル「k」からWレジスタを減算します(2の補数法)。結果はWレジスタに書き込まれます。

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W < 3:0 > > k < 3:0 >$
DC = 1	$W < 3:0 > \leq k < 3:0 >$

SUBWF Subtract W from f

構文: `[label] SUBWF f,d`
 オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
 動作: $(f) - (W) \rightarrow (\text{destination})$
 影響を受けるステータス: C, DC, Z
 説明: レジスタ「f」から、Wレジスタを減算します(2の補数法)。「d」が「0」の場合、結果はWレジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W < 3:0 > > f < 3:0 >$
DC = 1	$W < 3:0 > \leq f < 3:0 >$

SUBWFB Subtract W from f with Borrow

構文: `SUBWFB f{,d}`
 オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
 動作: $(f) - (W) - (\overline{B}) \rightarrow \text{dest}$
 影響を受けるステータス: C, DC, Z
 説明: レジスタ「f」からWレジスタとボローフラグ(キャリー)を減算します(2の補数法)。「d」が「0」の場合、結果はWレジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

PIC16(L)F1764/5/8/9

SWAPF Swap Nibbles in f

構文: [label] SWAPF f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)
影響を受けるステータス: なし
説明: レジスタ「f」の上位ニブルと下位ニブルを入れ換えます。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き込まれます。

TRIS Load TRIS Register with W

構文: [label] TRIS f
オペランド: $5 \leq f \leq 7$
動作: (W) → TRIS register 'f'
影響を受けるステータス: なし
説明: W レジスタから、TRIS レジスタにデータを移動します。
「f」= 5 の場合、TRISA にデータが書き込まれます。
「f」= 6 の場合、TRISB にデータが書き込まれます。
「f」= 7 の場合、TRISC にデータが書き込まれます。

XORLW Exclusive OR literal with W

構文: [label] XORLW k
オペランド: $0 \leq k \leq 255$
動作: (W).XOR.k → (W)
影響を受けるステータス: Z
説明: W レジスタの内容と 8 ビットのリテラル「k」を XOR 演算します。結果は W レジスタに書き込まれます。

XORWF Exclusive OR W with f

構文: [label] XORWF f,d
オペランド: $0 \leq f \leq 127$
 $d \in [0,1]$
動作: (W).XOR.(f) → (destination)
影響を受けるステータス: Z
説明: W レジスタの内容とレジスタ「f」を XOR 演算します。「d」が「0」の場合、結果は W レジスタに書き込まれます。「d」が「1」の場合、結果はレジスタ「f」に書き戻されます。

36.0 電氣的仕様

36.1 絶対最大定格 (†)

通電中の周囲温度 -40 ~ +125 °C

保管温度 -65 ~ +150 °C

V_{SS} を基準とするピン電圧

VDD ピン

PIC16F1764/5/8/9 -0.3 ~ +6.5 V

PIC16LF1764/5/8/9 -0.3 ~ +4.0 V

MCLR ピン -0.3 ~ +9.0 V

その他全てのピン -0.3 ~ +(V_{DD} + 0.3 V)

最大電流

V_{SS} ピン (1)

-40 °C ≤ T_A ≤ +85 °C 170 mA

-40 °C ≤ T_A ≤ +125 °C 70 mA

VDD ピン (1)

-40 °C ≤ T_A ≤ +85 °C 170 mA

-40 °C ≤ T_A ≤ +125 °C 70 mA

全ての I/O ピン ±25 mA

全ての最大電流 I/O ピン ±100 mA

クランプ電流、I_K (V ピン < 0 または V ピン > V_{DD}) ±20 mA

総消費電力 (2) 800 mW

Note 1: 最大電流定格では、I/O ピン間で負荷を均等に分散する必要があります。最大電流定格は、デバイスパッケージの消費電力特性によって制限される場合があります。表 36-6: 温度特性を参照してデバイス仕様値を求めます。

2: 消費電力は下式で求めます。

$$P_{dis} = V_{DD} * \{I_{dd} - \sum I_{oh}\} + \sum \{V_{DD} - V_{oh}\} * I_{oh} + \sum \{V_{ol}\} * I_{ol}$$

† 注意: ここに記載した「絶対最大定格」を超える条件は、デバイスに恒久的な損傷を生じる可能性があります。これはストレス定格です。本書の動作表に示す条件外でのデバイス運用は想定していません。長期間にわたる最大定格条件での動作や保管は、デバイスの信頼性に影響する可能性があります。

PIC16(L)F1764/5/8/9

36.2 標準動作条件

全てのデバイスの標準動作条件は以下のように定義されます。

動作電圧: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$

動作温度: $T_{A_MIN} \leq T_A \leq T_{A_MAX}$

V_{DD} — 動作電源電圧⁽¹⁾

PIC16LF1764/5/8/9

V_{DDMIN} (Fosc ≤ 16 MHz)..... +1.8 V

V_{DDMIN} (Fosc > 16 MHz)..... +2.5 V

V_{DDMAX} +3.6 V

PIC16F1764/5/8/9

V_{DDMIN} (Fosc ≤ 16 MHz)..... +2.3 V

V_{DDMIN} (> 16 MHz)..... +2.5 V

V_{DDMAX} +5.5 V

T_A - 動作時周囲温度レンジ

産業用温度レンジ

T_{A_MIN}..... -40 °C

T_{A_MAX}..... +85 °C

拡張温度レンジ

T_{A_MIN}..... -40 °C

T_{A_MAX}..... +125 °C

Note 1: 「DC 特性：電源電圧」表のパラメータ [D001](#) を参照してください。

図 36-1: 電圧と周波数、 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +125\text{ }^{\circ}\text{C}$ 、PIC16F1764/5/8/9 のみ

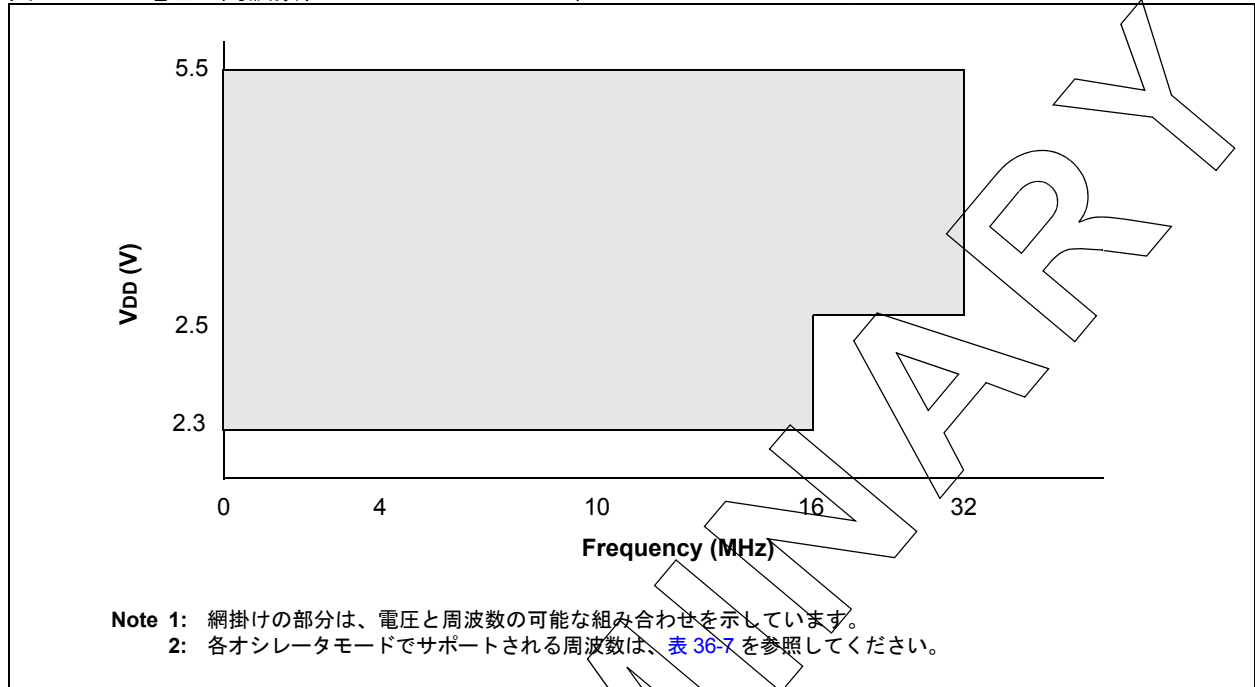
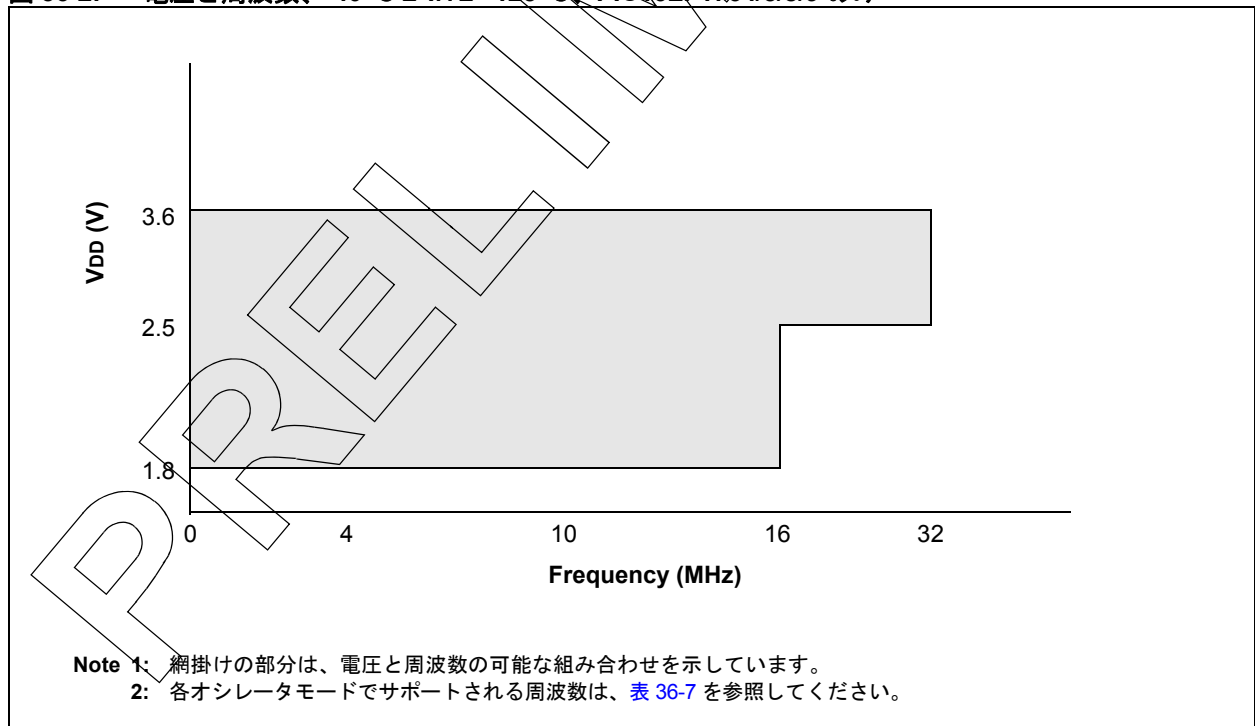


図 36-2: 電圧と周波数、 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +125\text{ }^{\circ}\text{C}$ 、PIC16LF1764/5/8/9 のみ



PIC16(L)F1764/5/8/9

36.3 DC 特性

表 36-1: 電源電圧

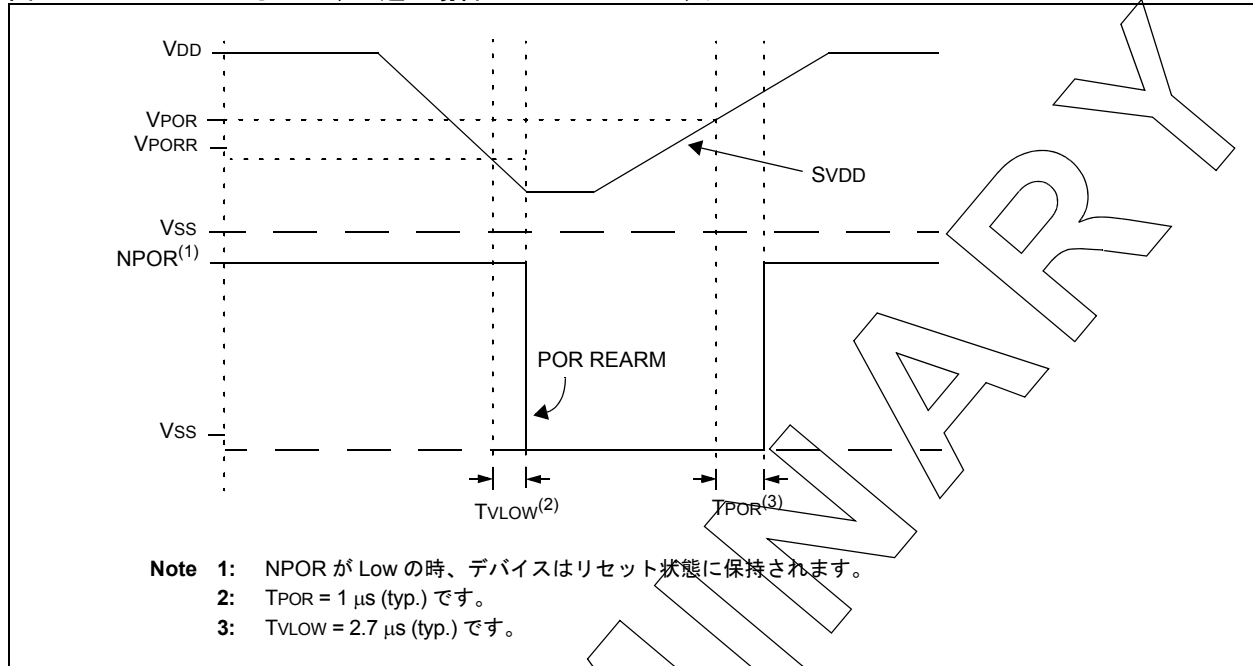
PIC16LF1764/5/8/9		標準動作条件 (特に明記しない場合)					
PIC16F1764/5/8/9							
パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
D001	VDD	電源電圧	VDDMIN	—	VDDMAX	V	Fosc ≤ 16 MHz Fosc ≤ 32 MHz (Note 2)
			1.8 2.5	—	3.6 3.6		
D001		PIC16F1764/5/8/9	2.3 2.5	—	5.5 5.5	V V	Fosc ≤ 16 MHz; Fosc ≤ 32 MHz (Note 2)
D002*	VDR	RAM データ保持電圧 ⁽¹⁾	1.5	—	—	V	デバイスがスリープの場合
			1.7	—	—	V	デバイスがスリープの場合
D002A*	VPOR	パワーオンリセットリリース電圧 ⁽³⁾	—	1.6	—	V	
			—	1.6	—	V	
D002B*	VPORR*	パワーオンリセットリアーム電圧 ⁽³⁾	—	0.8	—	V	
			—	1.5	—	V	
D003	VFVR	固定参照電圧値 ⁽⁴⁾	-4	—	+4	%	1x ゲイン、1.024、VDD ≥ 2.5 V、 -40 ~ +85 °C
			-4	—	+4	%	2x ゲイン、2.048、VDD ≥ 2.5 V、 -40 ~ +85 °C
			-5	—	+5	%	4x ゲイン、4.096、VDD ≥ 4.5 V、 -40 ~ +85 °C
D004*	SVDD	VDD 立ち上がり速度 ⁽²⁾	0.05	—	—	V/ms	パワーオンリセット信号の適切なリリースを確実にします。

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note**
- これは、スリープ中に VDD を下げた時に RAM データを失わない最小値です。
 - 32 MHz 動作には PLL が必要です。
 - 図 36-3: VDD の立ち上がりが遅い場合の POR と POR リアームを参照してください。
 - 産業用温度レンジのみ。

図 36-3: VDD の立ち上がりが遅い場合の POR と POR リアーム



PIC16(L)F1764/5/8/9

表 36-2: 消費電流 (IDD)^(1, 2)

PIC16LF1764/5/8/9		標準動作条件 (特に明記しない場合)					
PIC16F1764/5/8/9							
パラメータ No.	デバイス特性	最小	Typ.†	最大	単位	条件	
						VDD	Note
D009	LDO レギュレータ	—	75	—	μA	—	高消費電力モード、通常動作
		—	15	—	μA	—	スリープ、VREGCON<1> = 0
		—	0.3	—	μA	—	スリープ、VREGCON<1> = 1
D010		—	8	—	μA	1.8	FOSC = 32 kHz、 LP オシレータモード、 -40 °C ≤ TA ≤ +85 °C
		—	12	—	μA	3.0	
D010		—	15	—	μA	2.3	FOSC = 32 kHz、 LP オシレータモード (Note 4) -40 °C ≤ TA ≤ +85 °C
		—	17	—	μA	3.0	
		—	21	—	μA	5.0	
D012		—	140	—	μA	1.8	FOSC = 4 MHz、 XT オシレータモード
		—	250	—	μA	3.0	
D012		—	210	—	μA	2.3	FOSC = 4 MHz、 XT オシレータモード
		—	280	—	μA	3.0	
		—	340	—	μA	5.0	
D014		—	115	—	μA	1.8	FOSC = 4 MHz、 外部クロック (ECM)、 中消費電力モード
		—	210	—	μA	3.0	
D014		—	180	—	μA	2.3	FOSC = 4 MHz、 外部クロック (ECM)、 中消費電力モード
		—	240	—	μA	3.0	
		—	300	—	μA	5.0	
D015		—	2.1	—	mA	3.0	FOSC = 32 MHz、 外部クロック (ECH)、 高消費電力モード
		—	2.5	—	mA	3.6	
D015		—	2.1	—	mA	3.0	FOSC = 32 MHz、 外部クロック (ECH)、 高消費電力モード
		—	2.2	—	mA	5.0	

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note 1:** アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプルアップ、MCLR = VDD、WDT 無効
- 2:** 電源電流は、主に動作電圧と周波数の関数です。これ以外にも、I/O ピンの負荷とスイッチングレート、オシレータの種類、内部コードの実行パターン、温度等の要因で消費電流は変化します。
- 3:** EXTRC オシレータ構成の場合、REXT を流れる電流は含まれていません。この抵抗を流れる電流は、 $I_R = V_{DD}/2 R_{EXT}$ (mA) の式で求める事ができます (REXT の単位は kΩ)。
- 4:** FVR と BOR は無効です。
- 5:** 8 MHz の水晶振動子で 4x PLL 有効

PIC16(L)F1764/5/8/9

表 36-2: 消費電流 (IDD)^(1, 2) (続き)

PIC16LF1764/5/8/9		標準動作条件 (特に明記しない場合)					
PIC16F1764/5/8/9							
パラメータ No.	デバイス特性	最小	Typ.†	最大	単位	条件	
						VDD	Note
D017		—	130	—	μA	1.8	Fosc = 500 kHz、 MFINTOSC モード
		—	150	—	μA	3.0	
D017		—	150	—	μA	2.3	Fosc = 500 kHz、 MFINTOSC モード
		—	170	—	μA	3.0	
		—	220	—	μA	5.0	
D019		—	0.8	—	mA	1.8	Fosc = 16 MHz、 HFINTOSC モード
		—	1.2	—	mA	3.0	
D019		—	1.0	—	mA	2.3	Fosc = 16 MHz、 HFINTOSC モード
		—	1.3	—	mA	3.0	
		—	1.4	—	mA	5.0	
D020		—	2.1	—	mA	3.0	Fosc = 32 MHz、 HFINTOSC モード
		—	2.5	—	mA	3.6	
D020		—	2.1	—	mA	3.0	Fosc = 32 MHz、 HFINTOSC モード
		—	2.2	—	mA	5.0	
D022		—	2.1	—	mA	3.0	Fosc = 32 MHz、 HS オシレータモード (Note 5)
		—	2.5	—	mA	3.6	
D022		—	2.1	—	mA	3.0	Fosc = 32 MHz HS オシレータモード (Note 5)
		—	2.2	—	mA	5.0	

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note** 1: アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプルアップ、MCLR = VDD、WDT 無効
- 2: 電源電流は、主に動作電圧と周波数の関数です。これ以外にも、I/O ピンの負荷とスイッチングレート、オシレータの種類、内部コードの実行パターン、温度等の要因で消費電流は変化します。
- 3: EXTRC オシレータ構成の場合、REXT を流れる電流は含まれていません。この抵抗を流れる電流は、 $I_R = V_{DD} / 2 R_{EXT}$ (mA) の式で求める事ができます (REXT の単位は kΩ)。
- 4: FVR と BOR は無効です。
- 5: 8 MHz の水晶振動子で 4x PLL 有効

PIC16(L)F1764/5/8/9

表 36-3: 省電力時の電流 (IPD)^(1, 2)

PIC16LF1764/5/8/9		動作条件:(特に明記しない場合) 低消費電力スリープ						
PIC16F1764/5/8/9		低消費電力スリープ、VREGPM = 1						
パラメータ No.	デバイス特性	最小	Typ.†	最大 +85 °C	最大 +125 °C	単位	条件	
							VDD	Note
D023	ベース IPD	—	0.05	1.0	8.0	μA	1.8	WDT、BOR、FVR、SOSC は無効、周辺モジュールは全て非アクティブ
		—	0.08	2.0	9.0	μA	3.0	
D023	ベース IPD	—	0.3	3	11	μA	2.3	WDT、BOR、FVR、SOSC は無効、周辺モジュールは全て非アクティブ、低消費電力スリープ
		—	0.4	4	12	μA	3.0	
		—	0.5	6	15	μA	5.0	
D023A	ベース IPD	—	9.8	16	18	μA	2.3	WDT、BOR、FVR、SOSC は無効、周辺モジュールは全て非アクティブ、通常消費電力スリープ VREGPM = 0
		—	10.3	18	20	μA	3.0	
		—	11.5	21	26	μA	5.0	
D024		—	0.5	6	14	μA	1.8	WDT の電流
		—	0.8	7	17	μA	3.0	
D024		—	0.8	6	15	μA	2.3	WDT の電流
		—	0.9	7	20	μA	3.0	
		—	1.0	8	22	μA	5.0	
		—	1.0	8	22	μA	5.0	
D025		—	15	28	30	μA	1.8	FVR の電流
		—	18	30	33	μA	3.0	
D025		—	18	33	35	μA	2.3	FVR の電流
		—	19	35	37	μA	3.0	
		—	20	37	39	μA	5.0	
D026		—	7.5	25	28	μA	3.0	BOR の電流
D026		—	10	25	28	μA	3.0	BOR の電流
		—	12	28	31	μA	5.0	
D027		—	0.5	4	10	μA	3.0	LPBOR の電流
D027		—	0.8	6	14	μA	3.0	LPBOR の電流
		—	1	8	17	μA	5.0	
		—	1	8	17	μA	5.0	
D028		—	0.5	5	9	μA	1.8	SOSC の電流
		—	0.8	8.5	12	μA	3.0	
D028		—	1.1	6	10	μA	2.3	SOSC の電流
		—	1.3	8.5	20	μA	3.0	
		—	1.4	10	25	μA	5.0	
D029		—	0.05	2	9	μA	1.8	ADC の電流 (Note 3)、A/D 変換を実行中でない場合
		—	0.08	3	10	μA	3.0	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note 1:** 周辺モジュールの電流は、その周辺モジュールが有効時に消費する電流とベース IPD を合計したものです。周辺モジュールのみの Δ 電流は、この合計値からベース IDD または IPD 電流を減算して求める事ができます。消費電流の合計は最大値を使って求めます。
- 2:** スリープ中のパワーダウン電流は、オシレータの種類にかかわらず一定です。パワーダウン電流は、デバイスをスリープに移行し、全ての I/O ピンをハイインピーダンス状態にして V_{SS} に接続して計測しています。
- 3:** ADC クロック源は FRC です。

表 36-3: 省電力時の電流 (IPD)^(1, 2) (続き)

PIC16LF1764/5/8/9		動作条件:(特に明記しない場合) 低消費電力スリープ							
PIC16F1764/5/8/9		低消費電力スリープ、VREGPM = 1							
パラメータ No.	デバイス特性	最小	Typ.†	最大 +85 °C	最大 +125 °C	単位	条件		
							VDD	Note	
D029		—	0.3	4	12	μA	2.3	ADC の電流 (Note 3)、 A/D 変換を実行中でない場合	
		—	0.4	5	13	μA	3.0		
		—	0.5	7	16	μA	5.0		
D030		—	250	—	—	μA	1.8	ADC の電流 (Note 3)、 A/D 変換を実行中の場合	
		—	280	—	—	μA	3.0		
D030		—	230	—	—	μA	2.3	ADC の電流 (Note 3)、 A/D 変換を実行中の場合	
		—	250	—	—	μA	3.0		
		—	350	—	—	μA	5.0		
D031		—	250	650	—	μA	3.0	オペアンプ (高消費電力)	
D031		—	250	650	—	μA	3.0	オペアンプ (高消費電力)	
		—	350	650	—	μA	5.0		
D032		—	250	600	—	μA	1.8	コンパレータ	
		—	300	650	—	μA	3.0		
D032		—	280	600	—	μA	2.3	コンパレータ、 VREGPM = 0	
		—	300	650	—	μA	3.0		
		—	310	650	—	μA	5.0		

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °Cでの値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note** 1: 周辺モジュールの電流は、その周辺モジュールが有効時に消費する電流とベース IPD を合計したものです。周辺モジュールのみの Δ 電流は、この合計値からベース IDD または IPD 電流を減算して求める事ができます。消費電流の合計は最大値を使って求めます。
- 2: スリープ中のパワーダウン電流は、オシレータの種類にかかわらず一定です。パワーダウン電流は、デバイスをスリープに移行し、全ての I/O ピンをハイインピーダンス状態にして V_{SS} に接続して計測しています。
- 3: ADC クロック源は FRC です。

PIC16(L)F1764/5/8/9

表 36-4: I/O ポート

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
	V _{IL}	入力 Low 電圧					
		I/O ポート :					
D034		TTL バッファ使用	—	—	0.8	V	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
D034A			—	—	$0.15 V_{DD}$	V	$1.8\text{ V} \leq V_{DD} \leq 4.5\text{ V}$
D035		シュミットトリガ バッファ使用	—	—	$0.2 V_{DD}$	V	$2.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
		I ² C™ レベル	—	—	$0.3 V_{DD}$	V	
		SMBus レベル	—	—	0.8	V	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
D036		MCLR、OSC1 (EXTRC モード)	—	—	$0.2 V_{DD}$	V	(Note 1)
D036A		OSC1 (HS モード)	—	—	$0.3 V_{DD}$	V	
	V _{IH}	入力 High 電圧					
		I/O ポート :					
D040		TTL バッファ使用	2.0	—	—	V	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
D040A			$0.25 V_{DD} + 0.8$	—	—	V	$1.8\text{ V} \leq V_{DD} \leq 4.5\text{ V}$
D041		シュミットトリガ バッファ使用	$0.8 V_{DD}$	—	—	V	$2.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
		I ² C™ レベル	$0.7 V_{DD}$	—	—	V	
		SMBus レベル	2.1	—	—	V	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
D042		MCLR	$0.8 V_{DD}$	—	—	V	
D043A		OSC1 (HS モード)	$0.7 V_{DD}$	—	—	V	
D043B		OSC1 (EXTRC オシレータ)	$0.9 V_{DD}$	—	—	V	$V_{DD} \geq 2.0\text{ V}$ (Note 1)
	I _{IL}	入力リーク電流⁽²⁾					
D060		I/O ポート	—	±5	±125	nA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, 85 °C ハイインピーダンス状態のピン、85 °C
			—	±5	±1000	nA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, 85 °C ハイインピーダンス状態のピン、125 °C
D061		MCLR ⁽³⁾	—	±50	±200	nA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, 85 °C ハイインピーダンス状態のピン、85 °C
	IPUR	弱プルアップ電流					
D070*			25	100	200	μA	$V_{DD} = 3.3\text{ V}$, $V_{PIN} = V_{SS}$
			25	140	300	μA	$V_{DD} = 5.0\text{ V}$, $V_{PIN} = V_{SS}$
	V _{OL}	出力 Low 電圧⁽⁴⁾					
D080		標準 I/O ポート	—	—	0.6	V	I _{OL} = 8 mA, $V_{DD} = 5\text{ V}$ I _{OL} = 6 mA, $V_{DD} = 3.3\text{ V}$ I _{OL} = 1.8 mA, $V_{DD} = 1.8\text{ V}$
D080A		大電流駆動 I/O ポート	—	—	0.6	V	I _{OH} = 10 mA, $V_{DD} = 2.3\text{ V}$, $HIDC_x = 1$
			—	0.6	—	V	I _{OH} = 32 mA, $V_{DD} = 3.0\text{ V}$, $HIDC_x = 1$
			—	0.6	—	V	I _{OH} = 51 mA, $V_{DD} = 5.0\text{ V}$, $HIDC_x = 1$
	V _{OH}	出力 High 電圧⁽⁴⁾					
D090		標準 I/O ポート	$V_{DD} - 0.7$	—	—	V	I _{OH} = 3.5 mA, $V_{DD} = 5\text{ V}$ I _{OH} = 3 mA, $V_{DD} = 3.3\text{ V}$ I _{OH} = 1 mA, $V_{DD} = 1.8\text{ V}$
D090A		大電流駆動 I/O ポート	$V_{DD} - 0.7$	—	—	V	I _{OH} = 10 mA, $V_{DD} = 2.3\text{ V}$, $HIDC_x = 1$
			—	$V_{DD} - 0.7$	—	V	I _{OH} = 37 mA, $V_{DD} = 3.0\text{ V}$, $HIDC_x = 1$
			—	$V_{DD} - 0.7$	—	V	I _{OH} = 54 mA, $V_{DD} = 5.0\text{ V}$, $HIDC_x = 1$

* これらのパラメータは特性データであり、テストしていません。
† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C の値です。これらのパラメータは設計上の目安であり、テストしていません。
Note 1: EXTRC オシレータ構成では、OSC1/CLKIN ピンはシュミットトリガ入力です。EXTRC モードでは外部クロックの使用は推奨しません。
2: 負の電流値は、ピンからのソース電流として定義しています。
3: MCLR ピンのリーク電流は、印加する電圧レベルによって大きく変化します。仕様値のレベルは通常の動作条件での値を表します。入力電圧が異なると、より大きなリーク電流が計測される事があります。
4: CLKOUT モードの OSC2 を含みます。

表 36-4: I/O ポート (続き) (続き)

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
出力ピンに対する容量性負荷の仕様							
D101*	COSC2	OSC2 ピン	—	—	15	pF	XT、HS、LP モードで外部クロックを使って OSC1 を駆動している場合
D101A*	CIO	全ての I/O ピン	—	—	50	pF	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note**
- 1: EXTRC オシレータ構成では、OSC1/CLKIN ピンはシュミットトリガ入力です。EXTRC モードでは外部クロックの使用は推奨しません。
 - 2: 負の電流値は、ピンからのソース電流として定義しています。
 - 3: MCLR ピンのリーク電流は、印加する電圧レベルによって大きく変化します。仕様値のレベルは通常の動作条件での値を表します。入力電圧が異なると、より大きなリーク電流が計測される事があります。
 - 4: CLKOUT モードの OSC2 を含みます。

PRELIMINARY

PIC16(L)F1764/5/8/9

表 36-5: メモリのプログラミングに関する仕様

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
プログラムメモリのプログラミングに関する仕様							
D110	VIHH	MCLR/VPP ピンの電圧	8.0	—	9.0	V	(Note 2)
D111	IDDP	プログラム中の消費電流	—	—	10	mA	
D112	VBE	バルク消去用 VDD	2.7	—	VDDMAX	V	
D113	VPEW	行消去 / 書き込み用 VDD	VDDMIN	—	VDDMAX	V	
D114	IPPPGM	消去 / 書き込み時の MCLR/VPP の電流	—	1.0	—	mA	
D115	IDDPGM	消去 / 書き込み時の VDD の電流	—	5.0	—	mA	
プログラム フラッシュメモリ							
D121	EP	セル書き込み耐性	1 万	—	—	E/W	-40 °C ≤ TA ≤ +85 °C (Note 1)
D122	VPRW	読み書き用 VDD	VDDMIN	—	VDDMAX	V	
D123	TIW	自己書き込みサイクル時間	—	2	2.5	ms	
D124	TRETD	特性保持期間	—	40	—	年	他の仕様値に違反していない場合
D125	EHEFC	高書き換え耐性フラッシュセル	10 万	—	—	E/W	-0 °C ≤ TA ≤ +60 °C、 下位バイト、 最後の 128 アドレス

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

Note 1: 自己書き込みとブロック消去の場合

Note 2: 単電源プログラミングが無効の場合のみ必要

表 36-6: 温度特性

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	Typ.	単位	条件
TH01	θ_{JA}	接合部 - 大気間熱抵抗	70.0	°C/W	14 ピン PDIP パッケージ
			95.3	°C/W	14 ピン SOIC パッケージ
			100.0	°C/W	14 ピン TSSOP パッケージ
			51.5	°C/W	16 ピン QFN 4x4 mm パッケージ
			62.2	°C/W	20 ピン PDIP パッケージ
			87.3	°C/W	20 ピン SSOP
			77.7	°C/W	20 ピン SOIC パッケージ
			43.0	°C/W	20 ピン QFN 4x4 mm パッケージ
TH02	θ_{JC}	接合部 - ケース間熱抵抗	32.75	°C/W	14 ピン PDIP パッケージ
			31.0	°C/W	14 ピン SOIC パッケージ
			24.4	°C/W	14 ピン TSSOP パッケージ
			5.4	°C/W	16 ピン QFN 4x4 mm パッケージ
			27.5	°C/W	20 ピン PDIP パッケージ
			31.1	°C/W	20 ピン SSOP
			23.1	°C/W	20 ピン SOIC パッケージ
			5.3	°C/W	20 ピン QFN 4x4 mm パッケージ
TH03	T _{JMAX}	最高接合部温度	150	°C	
TH04	PD	消費電力	—	W	PD = P _{INTERNAL} + P _{I/O}
TH05	P _{INTERNAL}	内部消費電力	—	W	P _{INTERNAL} = I _{DD} x V _{DD} ⁽¹⁾
TH06	P _{I/O}	I/O 消費電力	—	W	P _{I/O} = $\Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P _{DER}	ディレーティング後電力	—	W	P _{DER} = P _D MAX (T _J - T _A)/ θ_{JA} ⁽²⁾

Note 1: I_{DD} は、出力ピンの負荷を駆動しないでチップのみを動作させた時の電流です。

2: T_A = 周囲温度、T_J = 接合部温度

PIC16(L)F1764/5/8/9

36.4 AC 特性

タイミングパラメータの記号は、以下のどちらかの書式で表します。

1. TppS2ppS
2. TppS

T		T	
F	周波数	T	時間
小文字 (pp) の種類と意味:			
pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	\overline{RD}
cs	\overline{CS}	rw	\overline{RD} または \overline{WR}
di	SDI	sc	SCK
do	SDO	ss	\overline{SS}
dt	データ入力	t0	T0CKI
io	I/O ポート	t1	T1CKI
mc	\overline{MCLR}	wr	\overline{WR}
大文字の種類と意味:			
S		P	
F	立ち下がり	P	周期
H	High	R	立ち上がり
I	不確定 (ハイインピーダンス)	V	確定
L	Low	Z	ハイインピーダンス

図 36-4: 負荷条件

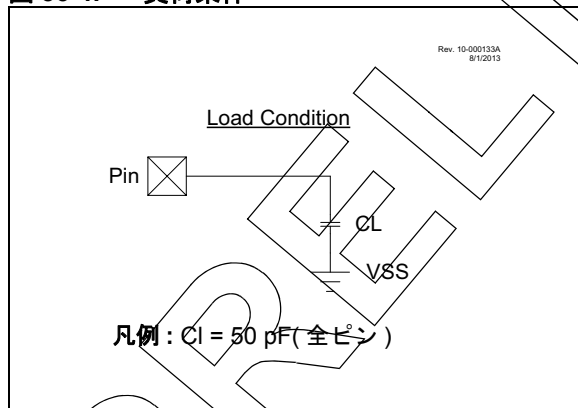


図 36-5: クロック タイミング

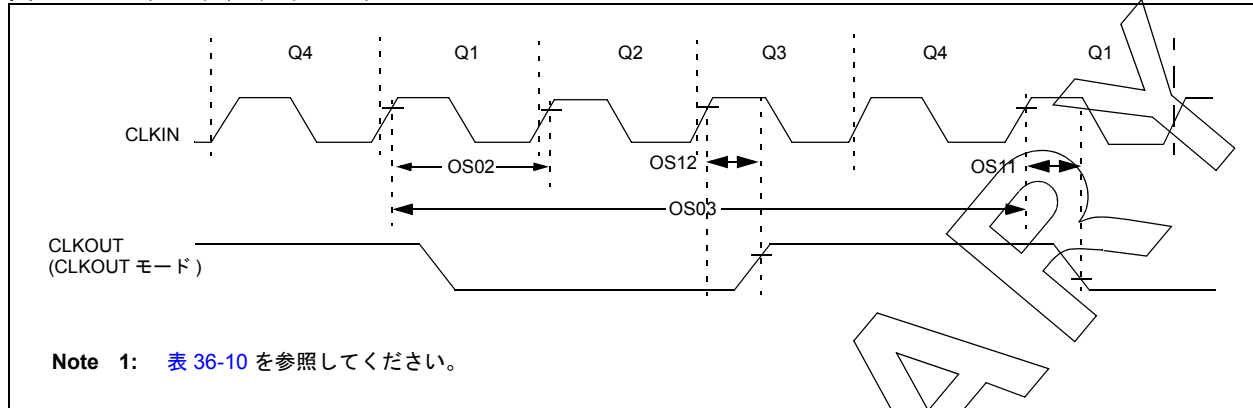


表 36-7: クロック オシレータのタイミング要件

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
OS01	Fosc	外部 CLKIN 周波数 (1)	DC	—	0.5	MHz	外部クロック (ECL)
			DC	—	4	MHz	外部クロック (ECM)
			DC	—	20	MHz	外部クロック (ECH)
	オシレータ周波数 (1)	—	32.768	—	kHz	LP オシレータ	
		0.1	—	4	MHz	XT オシレータ	
		1	—	4	MHz	HS オシレータ	
1		—	20	MHz	HS オシレータ、V _{DD} > 2.7 V		
DC	—	4	MHz	EXTRC、V _{DD} > 2.0 V			
OS02	Tosc	外部 CLKIN 周期 (1)	27	—	∞	μs	LP オシレータ
			250	—	∞	ns	XT オシレータ
			50	—	∞	ns	HS オシレータ
			50	—	∞	ns	外部クロック (EC)
	オシレータ周期 (1)	—	30.5	—	μs	LP オシレータ	
		250	—	10,000	ns	XT オシレータ	
50		—	1,000	ns	HS オシレータ		
250		—	—	ns	EXTRC		
OS03	Tcy	命令サイクル時間 (1)	125	Tcy	DC	ns	Tcy = 4/Fosc
OS04*	TosH、 TosL	外部 CLKIN High、 外部 CLKIN Low	2	—	—	μs	LP オシレータ
			100	—	—	ns	XT オシレータ
			20	—	—	ns	HS オシレータ
OS05*	TosR、 TosF	外部 CLKIN 立ち上がり、 外部 CLKIN 立ち下がり	0	—	∞	ns	LP オシレータ
			0	—	∞	ns	XT オシレータ
			0	—	∞	ns	HS オシレータ

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

Note 1: 命令サイクル周期 (Tcy) は、入力オシレータのタイムベース周期の 4 倍です。全ての仕様値は、標準動作条件下でデバイスにコードを実行させた時の、特定オシレータタイプの特性データに基づいています。これらの仕様の制限値を超えるとオシレータの動作が不安定になったり、予測以上の電流を消費する事があります。全てのデバイスは、OSC1 ピンに外部クロックを入力した状態で「最小値」で動作をテストしています。外部クロック入力を使った場合のサイクル時間の最大値は、全てのデバイスで「DC」(クロックなし)です。

PIC16(L)F1764/5/8/9

表 36-8: オシレータのパラメータ

標準動作条件 (特に明記しない場合)								
パラメータ No.	記号	特性	周波数許容範囲	最小	Typ.†	最大	単位	条件
OS08	HFosc	内部校正済み HFINTOSC 周波数 ⁽¹⁾	±2%	—	16.0	—	MHz	V _{DD} = 3.0 V、T _A = 25 °C (Note 2)
OS08A	MFosc	内部校正済み MFINTOSC 周波数 ⁽¹⁾	±2%	—	500	—	kHz	V _{DD} = 3.0 V、T _A = 25 °C (Note 2)
OS09	LFosc	内部 LFINTOSC 周波数	—	—	31	—	kHz	-40 °C ≤ T _A ≤ +125 °C
OS10*	T _{IOsc ST}	HFINTOSC	—	—	3.2	8	μs	
		スリープから復帰時の起動時間	—	—	24	35	μs	
OS10A*	T _{LFosc ST}	LFINTOSC	—	—	0.5	—	ms	-40 °C ≤ T _A ≤ +125 °C
		スリープから復帰時の起動時間	—	—	—	—	—	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note 1:** これらのオシレータ周波数の許容誤差を確保するには、デバイスのなるべく近くで V_{DD} と V_{SS} にデカップリングコンデンサを接続する必要があります。0.1 μF のコンデンサと 0.01 μF のコンデンサを並列に接続する事を推奨します。
- 2:** 図 36-6: デバイス V_{DD} と温度に対する HFINTOSC の周波数精度、
図 37-22: スリープ、HFINTOSC 信号源による復帰期間、PIC16LF1764/5/8/9 のみ、
図 36-6: デバイス V_{DD} と温度に対する HFINTOSC の周波数精度を参照してください。
- 3:** 図 37-20: V_{DD} と温度に対する LFINTOSC の周波数特性、PIC16LF1764/5/8/9 のみ、
図 37-21: V_{DD} と温度に対する LFINTOSC の周波数特性、PIC16F1764/5/8/9 のみを参照してください。

図 36-6: デバイス V_{DD} と温度に対する HFINTOSC の周波数精度

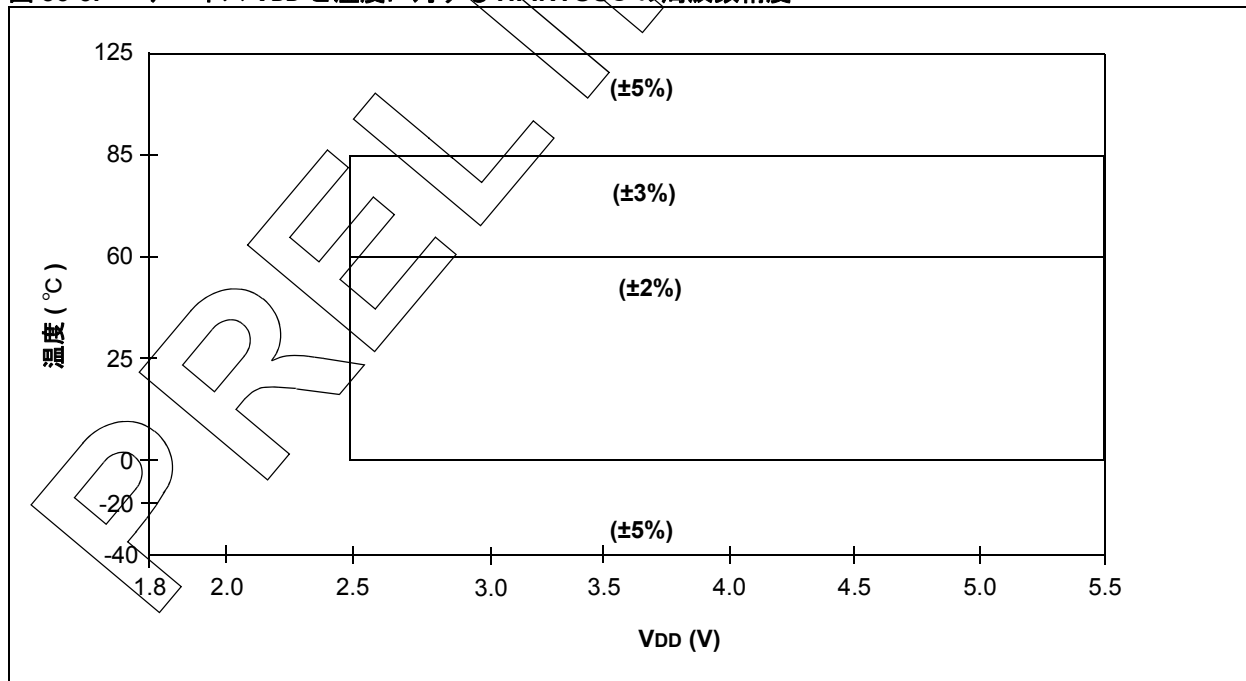


表 36-9: PLL クロック タイミング仕様

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
F10	FOSC	オシレータ周波数レンジ	4	—	8	MHz	
F11	FSYS	内蔵 VCO システム周波数	16	—	32	MHz	
F12	TRC	PLL 起動時間 (ロック時間)	—	—	2	ms	
F13*	ΔCLK	CLKOUT 安定性 (ジッタ)	-0.25%	—	+0.25%	%	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 5 V、25 °Cでの値です。これらのパラメータは設計上の目安であり、テストしていません。

PRELIMINARY

PIC16(L)F1764/5/8/9

図 36-7: CLKOUT と I/O タイミング

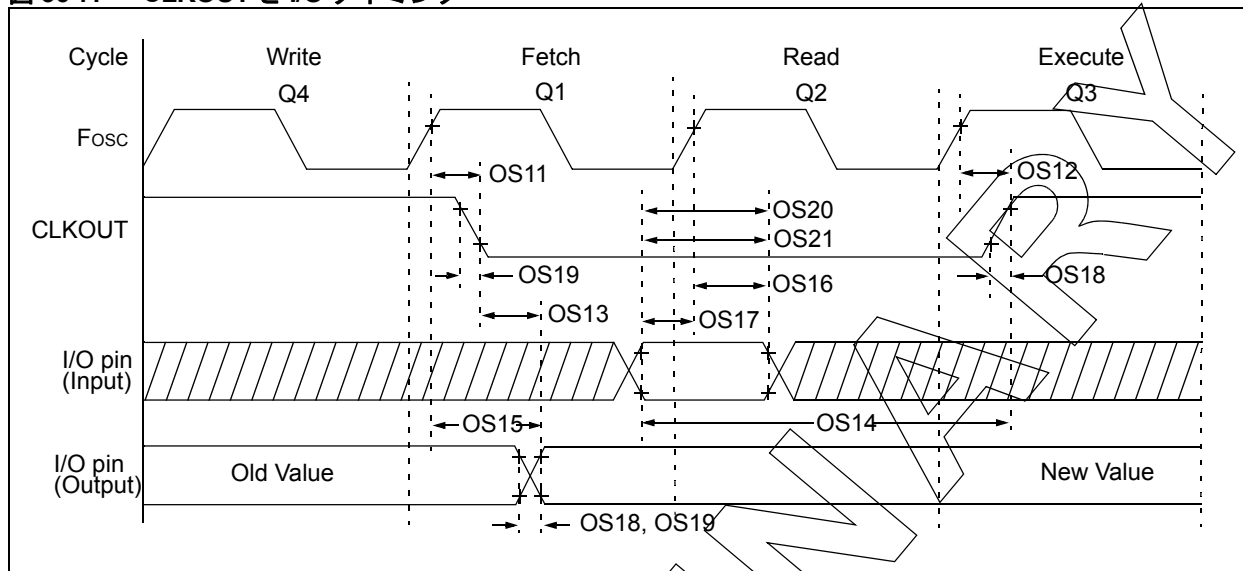


表 36-10: CLKOUT と I/O のタイミングパラメータ

標準動作条件 (特に明記しない場合)

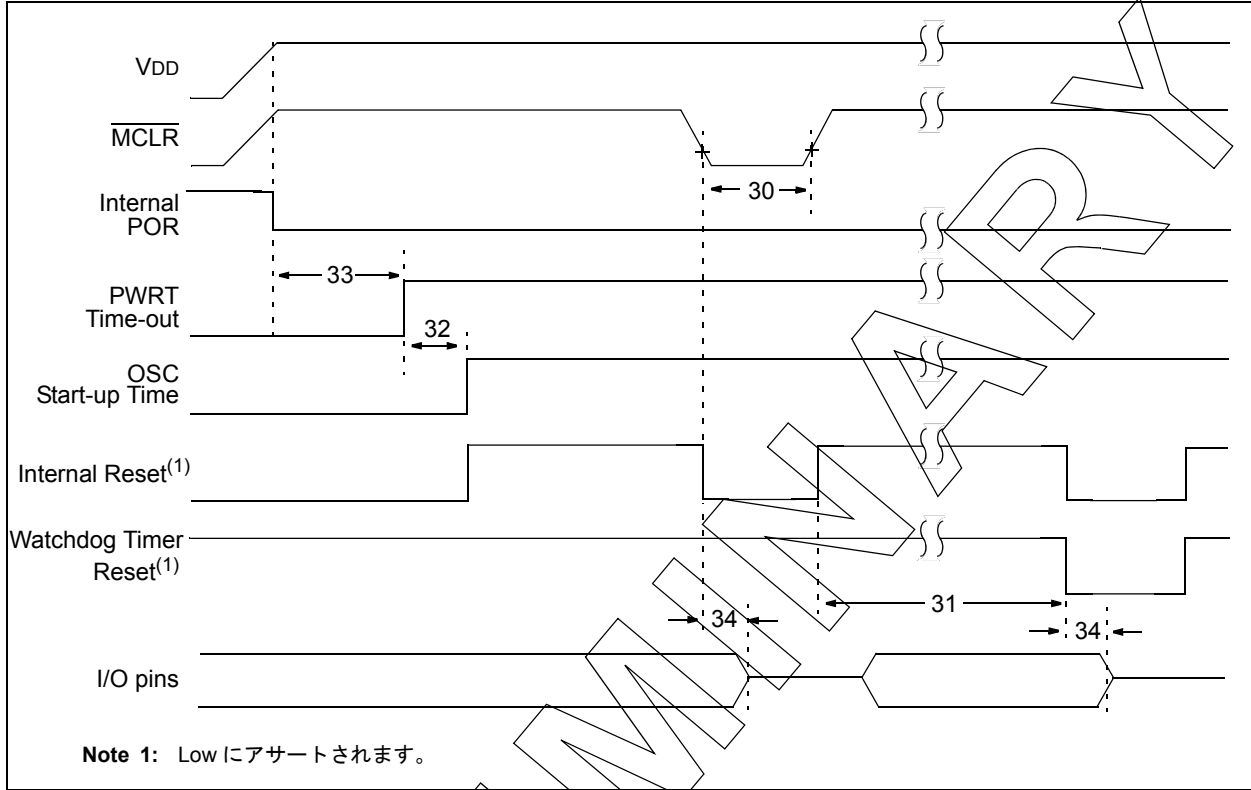
パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
OS11	TosH2ckL	Fosc から CLKOUT↓ までの時間 ⁽¹⁾	—	—	70	ns	3.3 V ≤ VDD ≤ 5.0 V
OS12	TosH2ckH	Fosc から CLKOUT までの時間 ⁽¹⁾	—	—	72	ns	3.3 V ≤ VDD ≤ 5.0 V
OS13	TckL2ioV	CLKOUT↓ からポート出力確定までの時間 ⁽¹⁾	—	—	20	ns	
OS14	TioV2ckH	ポート入力確定から CLKOUT までの時間 ⁽¹⁾	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc (Q1 サイクル) からポート出力確定までの時間	—	50	70*	ns	3.3 V ≤ VDD ≤ 5.0 V
OS16	TosH2iol	Fosc (Q2 サイクル) からポート入力無効までの時間 (I/O 入力ホールド時間)	50	—	—	ns	3.3 V ≤ VDD ≤ 5.0 V
OS17	TioV2osH	ポート入力確定から Fosc (Q2 サイクル) までの時間 (I/O 入力セットアップ時間)	20	—	—	ns	
OS18*	TioR	ポート出力立ち上がり時間	—	40 15	72 32	ns	VDD = 1.8 V 3.3 V ≤ VDD ≤ 5.0 V
OS19*	TioF	ポート出力立ち下がり時間	—	28 15	55 30	ns	VDD = 1.8 V 3.3 V ≤ VDD ≤ 5.0 V
OS20*	Tinp	INT ピン入力の High または Low 時間	25	—	—	ns	
OS21*	Tioc	状態変化割り込みの新しい入力レベル時間	25	—	—	ns	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V、25 °C での値です。

Note 1: EXTRC モードで CLKOUT 出力を 4 x Tosc として計測しています。

図 36-8: リセット、ウォッチドッグタイマ、オシレータ起動タイマ、パワーアップタイマのタイミング



PIC16(L)F1764/5/8/9

表 36-11: リセット、ウォッチドッグ タイマ、オシレータ起動タイマ、パワーアップ タイマ、ブラウンアウト リセットに関するパラメータ

標準動作条件 (特に明記しない場合)							
パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
30	TMCL	MCLR パルス幅 (Low)	2	—	—	μs	
31	TWDTLP	低消費電力ウォッチドッグ タイマ タイムアウト周期	10	16	27	ms	VDD = 3.3 ~ 5V 1:16 ブリスケーラ使用
32	TOST	オシレータ起動タイマ時間 ⁽¹⁾	—	1024	—	Tosc	
33*	TPWRT	パワーアップ タイマ時間、 PWRTE = 0	40	65	140	ms	
34*	TIOZ	MCLR Low または ウォッチドッグ タイマリセットから I/O ハイインピーダンス	—	—	2.0	μs	
35	VBOR	ブラウンアウト リセット電圧 ⁽²⁾	2.55	2.70	2.85	V	BORV = 0
			2.30	2.45	2.60	V	BORV = 1 (PIC16F1764/5/8/9)
			1.80	1.90	2.10	V	BORV = 1 (PIC16LF1764/5/8/9)
35A	VLPBOR	低消費電力ブラウンアウト	1.8	2.1	2.5	V	LPBOR = 1
36*	VHYST	ブラウンアウト リセットの ヒステリシス	0	25	75	mV	-40 °C ≤ TA ≤ +85 °C
37*	TBORDC	ブラウンアウト リセットの DC 応答 時間	1	3	35	μs	VDD ≤ VBOR

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

Note 1: オシレータ起動タイマ (OST) は、周波数にかかわらず最初の 1024 サイクルをカウントします。

2: これらの電圧の許容誤差を確保するには、デバイスのなるべく近くで VDD と VSS にデカップリング コンデンサを接続する必要があります。0.1 μF のコンデンサと 0.01 μF のコンデンサを並列に接続する事を推奨します。

図 36-9: Timer0 と Timer1 の外部クロックのタイミング

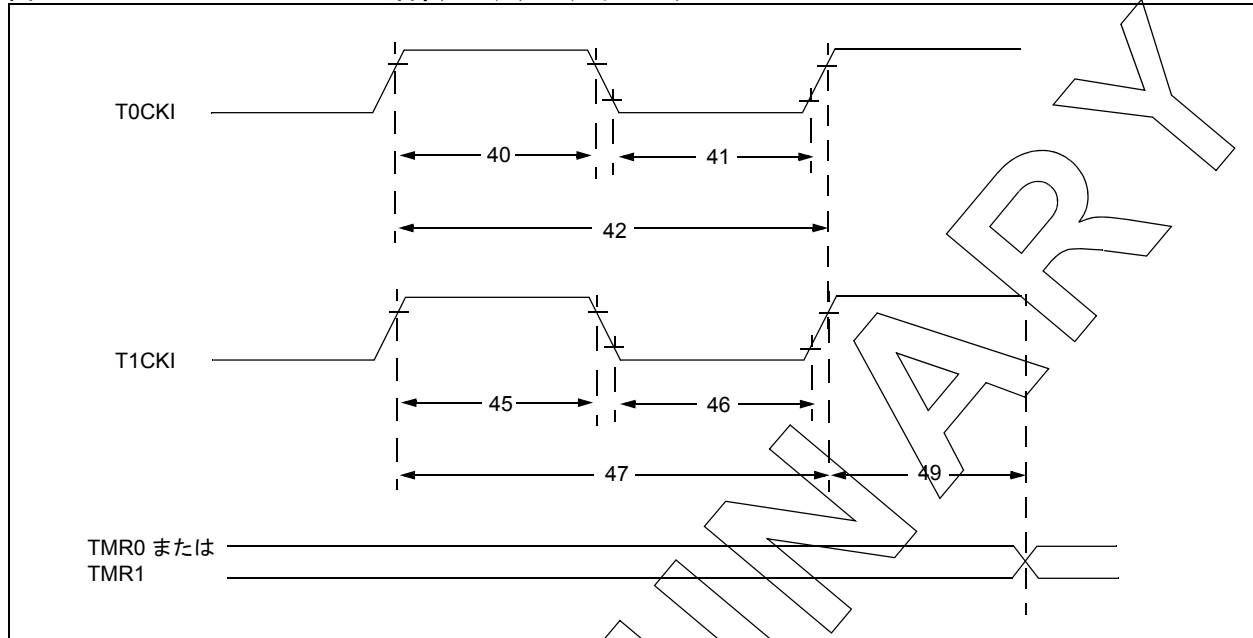
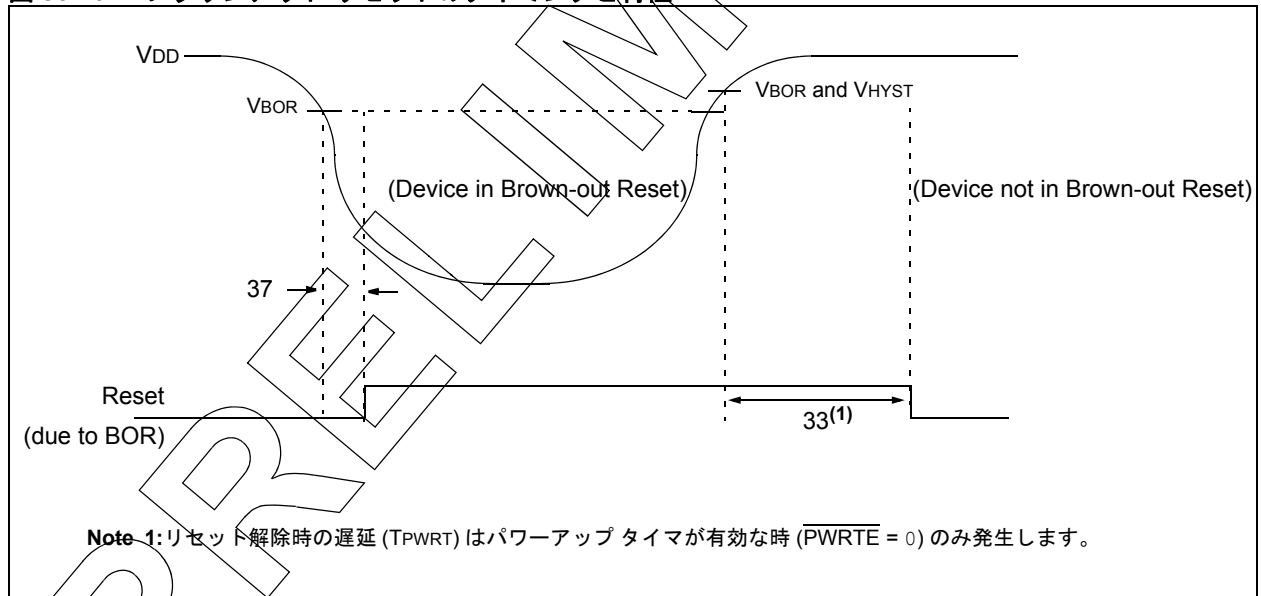


図 36-10: ブラウンアウトリセットのタイミングと特性



PIC16(L)F1764/5/8/9

表 36-12: Timer0 と Timer1 の外部クロック要件

標準動作条件 (特に明記しない場合)
動作温度 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +125\text{ }^{\circ}\text{C}$

パラメータ No.	記号	特性		最小	Typ.†	最大	単位	条件
40*	Tt0H	T0CKI パルス幅 (High)	プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			プリスケールあり	10	—	—	ns	
41*	Tt0L	T0CKI パルス幅 (Low)	プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			プリスケールあり	10	—	—	ns	
42*	Tt0P	T0CKI 周期		以下のどちらか大きい方: 20 または $T_{CY} + 40$ N	—	—	ns	N = プリスケール値
45*	Tt1H	T1CKI High 時間	同期、プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			同期、プリスケールあり	15	—	—	ns	
			非同期	30	—	—	ns	
46*	Tt1L	T1CKI Low 時間	同期、プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			同期、プリスケールあり	15	—	—	ns	
			非同期	30	—	—	ns	
47*	Tt1P	T1CKI 入力 周期	同期	以下のどちらか大きい方: 30 または $T_{CY} + 40$ N	—	—	ns	N = プリスケール値
			非同期	60	—	—	ns	
48	Ft1	セカンダリ オシレータ入力周波数レンジ (T1OSCEN ビットをセットしてオシレータを有効にした場合)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	外部クロックエッジからタイマのインクリメントまでの遅延		2 TOSC	—	7 TOSC	—	タイマの動作は同期モード

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは $3.0\text{ V}/25\text{ }^{\circ}\text{C}$ での値です。これらのパラメータは設計上の目安であり、テストしていません。

図 36-11: キャプチャ/コンペア/PWMのタイミング (CCP)

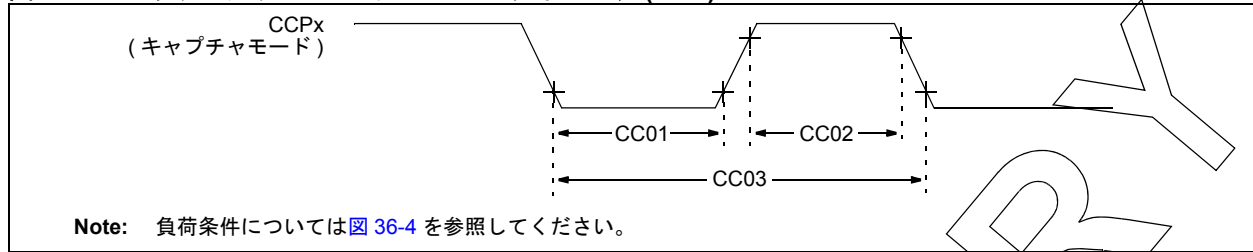


表 36-13: キャプチャ/コンペア/PWMの要件 (CCP)

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性		最小	Typ.†	最大	単位	条件
CC01*	TccL	CCPx 入力 Low 時間	プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			プリスケールあり	20	—	—	ns	
CC02*	TccH	CCPx 入力 High 時間	プリスケールなし	$0.5 T_{CY} + 20$	—	—	ns	
			プリスケールあり	20	—	—	ns	
CC03*	TccP	CCPx 入力周期		$\frac{3 T_{CY} + 40}{N}$	—	—	ns	N = プリスケール値

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C の値です。これらのパラメータは設計上の目安であり、テストしていません。

PIC16(L)F1764/5/8/9

図 36-12: CLC 伝播タイミング

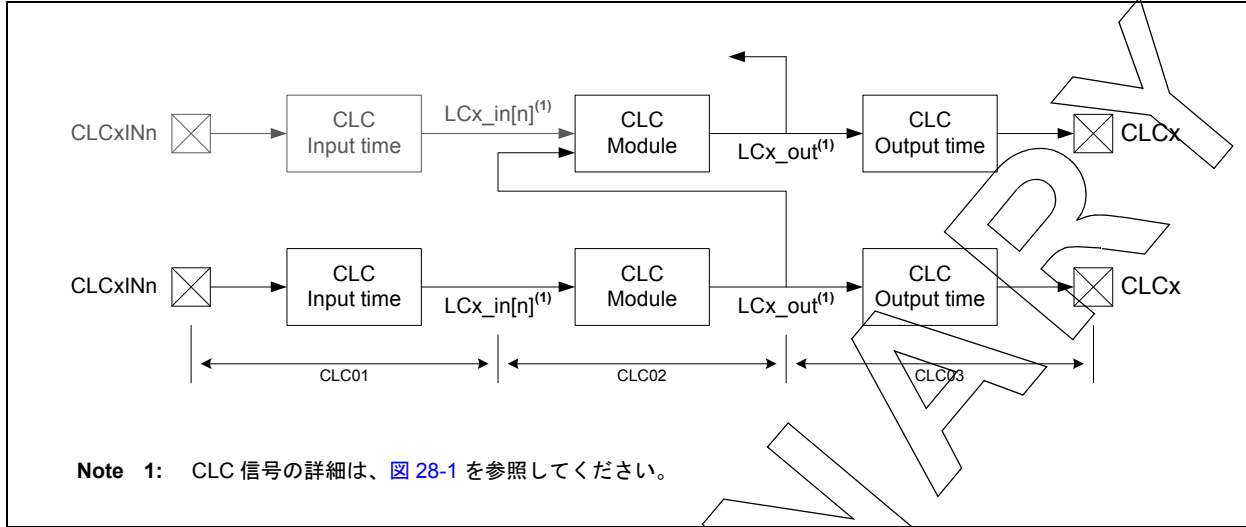


表 36-14: 構成可能なロジックセル (CLC) 特性

標準動作条件 (特に明記しない場合)
動作温度 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +125\text{ }^{\circ}\text{C}$

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
CLC01*	TCLCIN	CLC 入力時間	—	7	OS17	ns	(Note 1)
CLC02*	TCLC	CLC モジュールの入力から出力までの伝播時間	—	24	—	ns	$V_{DD} = 1.8\text{ V}$
			—	12	—	ns	$V_{DD} > 3.6\text{ V}$
CLC03*	TCLCOUT	CLC 出力時間	—	OS18	—	—	(Note 1)
			—	OS19	—	—	(Note 1)
CLC04*	FCLCMAX	CLC 最大スイッチング周波数	—	45	—	MHz	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは $3.0\text{ V}/25\text{ }^{\circ}\text{C}$ での値です。これらのパラメータは設計上の目安であり、テストしていません。

Note 1: OS17、OS18、OS19 の立ち上がりおよび立ち下がり時間は、表 36-10 を参照してください。

表 36-15: ADC (A/D コンバータ) の特性 (1、2、3、4)

動作条件 (特に明記しない場合) VDD = 3.0 V、TA = 25 °C、シングルエンド、2 μs TAD、VREF+ = 3 V、VREF- = VSS							
パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
AD01	NR	分解能	—	—	10	bit	
AD02	EIL	積分直線性誤差	—	—	±1.7	LSb	VREF = 3.0 V
AD03	EDL	微分直線性誤差	—	—	±1	LSb	ノー ミッシングコード、VREF = 3.0 V
AD04	EOFF	オフセット誤差	—	—	±2.5	LSb	VREF = 3.0 V
AD05	EGN	ゲインエラー	—	—	±2.0	LSb	VREF = 3.0 V
AD06	VREF	参照電圧	1.8	—	VDD	V	VREF = (VREF+ minus VREF-)
AD07	VAIN	フルスケール レンジ	VSS	—	VREF	V	
AD08	ZAIN	アナログ電圧源の推奨インピーダンス	—	—	10	kΩ	入力ピンに外付けの 0.01 μF コンデンサを使った場合、さらに高い値が可能

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note 1:** 総絶対誤差には、積分非直線性誤差、微分直線性誤差、オフセット誤差、ゲインエラーが含まれます。
2: 入力電圧が増大して A/D 変換結果が減少する事はありません。また、ミッシングコードもありません。
3: ADC VREF は、外部 VREF+ ピン、VDD ピン、FVR のうち参照入力として選択したものです。
4: 動作特性については、[セクション 37.0 「DC および AC 特性の図表」](#) を参照してください。

表 36-16: A/D 変換の要件

標準動作条件 (特に明記しない場合)							
パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
AD130*	TAD	ADC クロック周期 (TADC)	1.0	—	9.0	μs	Fosc に基づく
		ADC 内部 FRC オシレータ周期 (TFRC)	1.0	2.5	6.0	μs	ADCS<1:0> = 11 (ADC FRC モード)
AD131	Tcnv	変換時間 (アキュイジション時間を除く) ⁽¹⁾	—	13	—	TAD	GO/DONE ビットがセットされてから変換完了までの時間
AD132*	TACQ	アキュイジション時間	—	5.0	—	μs	
AD133*	THCD	ホールド コンデンサ切断時間	—	1/2 TAD	—		ADCS<2:0> = x11 (Fosc に基づく)
			—	1/2 TAD + 1 Tcy	—		ADCS<2:0> = x11 (FRC に基づく)

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

- Note 1:** ADRES レジスタは次の Tcy サイクルで読み出し可能です。

PIC16(L)F1764/5/8/9

図 36-13: A/D 変換タイミング (ADC クロック Fosc に基づく)

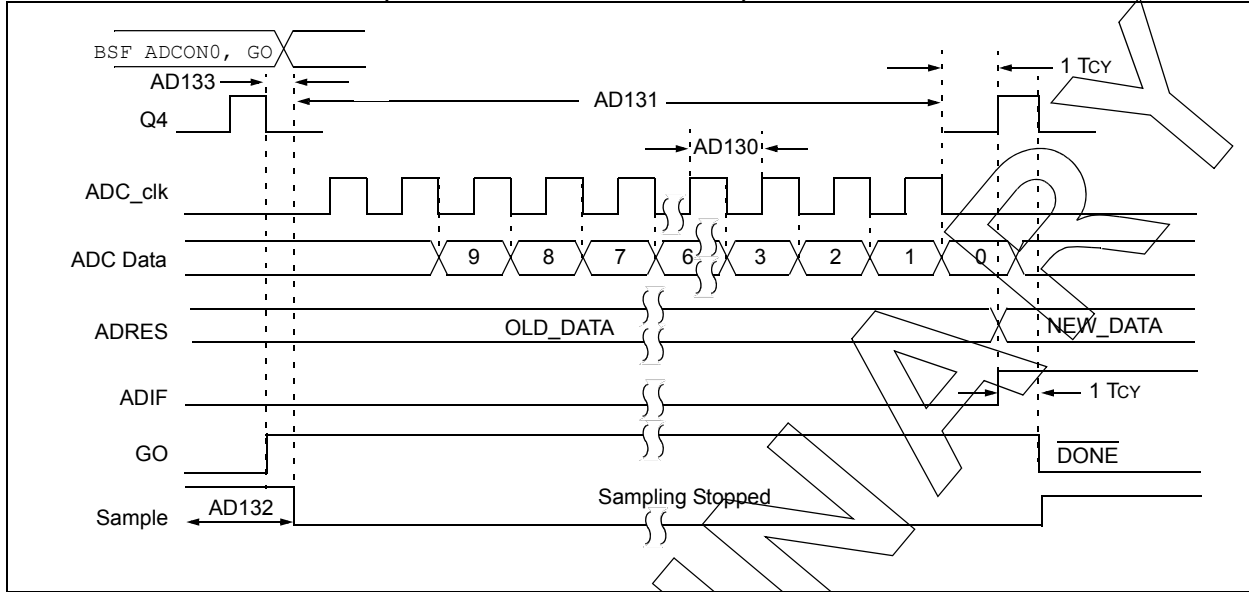


図 36-14: A/D 変換タイミング (FRC からの ADC クロック)

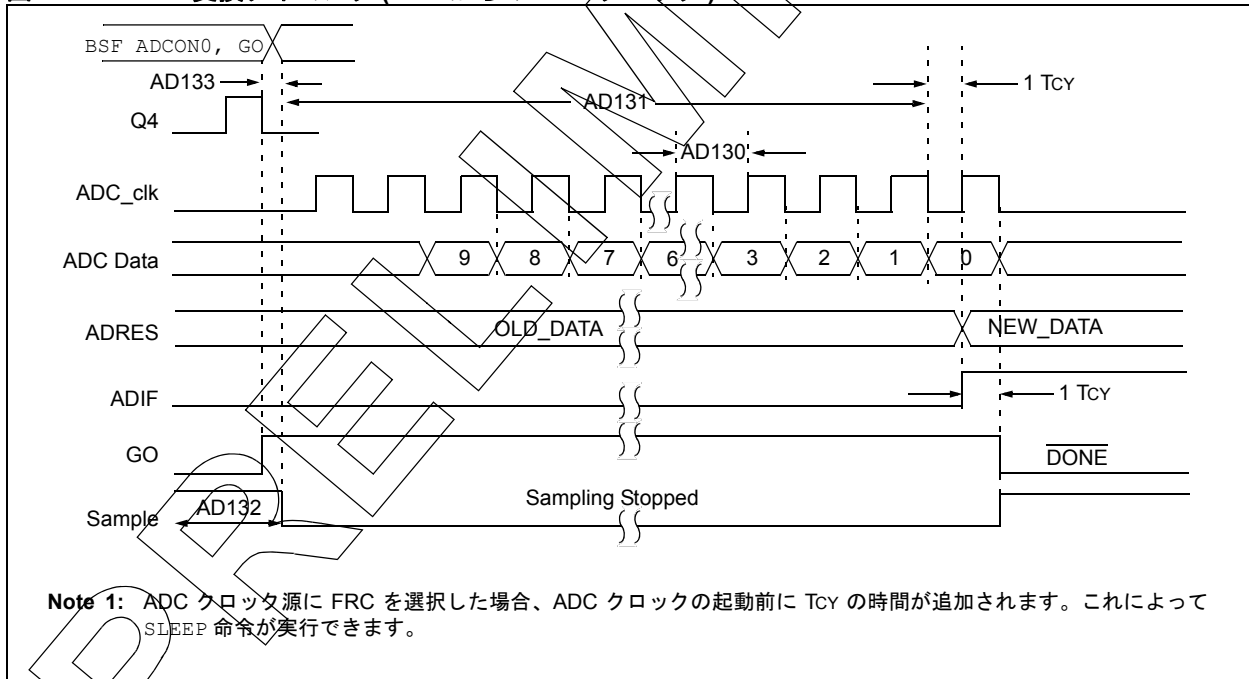


表 36-17: オペアンプ (OPA)

動作条件 (特に明記しない場合)
 $V_{DD} = 3.0\text{ V}$ 、 $T_A = 25\text{ }^\circ\text{C}$ 、 $OPAxSP = 1$ (高 GBWP モード)

パラメータ No.	記号	パラメータ	最小	Typ.	最大	単位	条件
OPA01*	GBWP	ゲイン帯域幅積	—	2	—	MHz	
OPA02*	TON	ターンオン時間	—	10	—	μs	
OPA03*	PM	位相マージン	—	40	—	°	
OPA04*	SR	スルーレート	—	3	—	$\text{V}/\mu\text{s}$	
OPA05	OFF	オフセット	—	± 3	± 9	mV	
OPA06	CMRR	コモンモード除去比	52	70	—	dB	
OPA07*	AOL	開ループゲイン	—	90	—	dB	
OPA08	VICM	入力コモンモード電圧	0	—	V_{DD}	V	$V_{DD} > 2.5\text{ V}$
OPA09*	PSRR	電源電圧変動除去比	—	80	—	dB	
OPA10*	HZ	ハイインピーダンス オン/オフ 時間	—	50	—	ns	

* これらのパラメータは特性データであり、テストしていません。

表 36-18: プログラマブル ランプ ジェネレータ (PRG) の仕様

動作条件 (特に明記しない場合)
 $V_{DD} = 3.0\text{ V}$ 、 $T_A = 25\text{ }^\circ\text{C}$ (特に明記のない場合)

パラメータ No.	記号	特性	最小	Typ.	最大	単位	注釈
PRG01	RRR	立ち上がりランプレート	—	1	—	$\text{V}/\mu\text{s}$	$\text{PRGxCON2} = 10\text{h}$
PRG02	FRR	立ち下がりランプレート	—	1	—	$\text{V}/\mu\text{s}$	$\text{PRGxCON2} = 10\text{h}$

* これらのパラメータは特性データであり、テストしていません。

Note 1: 応答時間は、コンパレータの一方の入力を $V_{DD}/2$ とし、もう一方の入力を V_{SS} から V_{DD} に遷移させて計測しています。

表 36-19: コンパレータの仕様

動作条件 (特に明記しない場合)
 $V_{DD} = 3.0\text{ V}$ 、 $T_A = 25\text{ }^\circ\text{C}$
 動作特性については、[セクション 37.0 「DC および AC 特性の図表」](#) を参照してください。

パラメータ No.	記号	特性	最小	Typ.	最大	単位	注釈
CM01	V_{IQFF}	入力オフセット電圧	—	± 2.5	± 5	mV	$V_{ICM} = V_{DD}/2$
CM02	V_{ICM}	入力コモンモード電圧	0	—	V_{DD}	V	
CM03	CMRR	コモンモード除去比	40	50	—	dB	
CM04A	TRESP ⁽¹⁾	応答時間 (立ち上がりエッジ)	—	60	85	ns	
CM04B		応答時間 (立ち下がりエッジ)	—	60	90	ns	
CM05*	TMC2OV	コンパレータのモード変更から出力確定までの時間*	—	—	10	μs	
CM06	CHYSTER	コンパレータ ヒステリシス	20	45	75	mV	$CxHYS = 1$

* これらのパラメータは特性データであり、テストしていません。

Note 1: 応答時間は、コンパレータの一方の入力を $V_{DD}/2$ とし、もう一方の入力を V_{SS} から V_{DD} に遷移させて計測しています。

PIC16(L)F1764/5/8/9

表 36-20: 10 ビット D/A コンバータ (DAC) の仕様

動作条件 (特に明記しない場合) VDD = 3.0 V、TA = 25 °C 動作特性については、セクション 37.0 「DC および AC 特性の図表」を参照してください。							
パラメータ No.	記号	特性	最小	Typ.	最大	単位	注釈
DAC01*	CLSB	ステップサイズ	—	VDD/1024	—	V	
DAC02	CINL	積分非直線性誤差 (2)	—	—	±1.5	LSb	0x004 ~ 0x3FB のコードに対して
DAC03	CDNL	微分非直線性誤差 (2)	—	—	±1	LSb	
DAC04	COFF	オフセット誤差 (2)	—	—	±3	LSb	
DAC05	CGN	ゲイン誤差 (2)	—	—	±3	LSb	
DAC06*	CR	単位抵抗値 (R)	—	300	—	Ω	
DAC07*	CST	セトリングタイム (1)	—	—	10	μs	

* これらのパラメータは特性データであり、テストしていません。

Note 1: セトリングタイムは、DACR<9:0> を「0x000」から「0x3FF」へ遷移させて計測しています。

2: ユニティゲインのオペアンプでバッファした場合です。

表 36-21: 5 ビット D/A コンバータ (DAC) の仕様

動作条件 (特に明記しない場合) VDD = 3.0 V、TA = 25 °C 動作特性については、セクション 37.0 「DC および AC 特性の図表」を参照してください。							
パラメータ No.	記号	特性	最小	Typ.	最大	単位	注釈
DAC10*	CLSB	ステップサイズ	—	VDD/32	—	V	
DAC11	CACC	絶対精度 (2)	—	—	±0.5	LSb	
DAC12*	CR	単位抵抗値 (R)	—	6000	—	Ω	
DAC13*	CST	セトリングタイム (1)	—	—	10	μs	

* これらのパラメータは特性データであり、テストしていません。

Note 1: セトリングタイムは、DACR<4:0> を「0x00」から「0x1F」へ遷移させて計測しています。

2: ユニティゲインのオペアンプでバッファした場合です。

表 36-22: ゼロクロス検出ピンの仕様

動作条件 (特に明記しない場合) VDD = 3.0 V、TA = 25 °C							
パラメータ No.	記号	特性	最小	Typ.	最大	単位	注釈
ZC01	ZCPINV	ゼロクロスピンの電圧	—	0.75	—	V	
ZC02	ZCSRC	ソース電流	—	-300	-600	μA	
ZC03	ZCSNK	シンク電流	—	300	600	μA	
ZC04	ZCISW	応答時間 (立ち上がりエッジ)	—	1	—	μs	
		応答時間 (立ち下がりエッジ)	—	1	—	μs	
ZC05	ZCOUT	応答時間 (立ち上がりエッジ)	—	1	—	μs	
		応答時間 (立ち下がりエッジ)	—	1	—	μs	

* これらのパラメータは特性データであり、テストしていません。

図 36-15: EUSART 同期送信 (マスタ/スレーブ) のタイミング

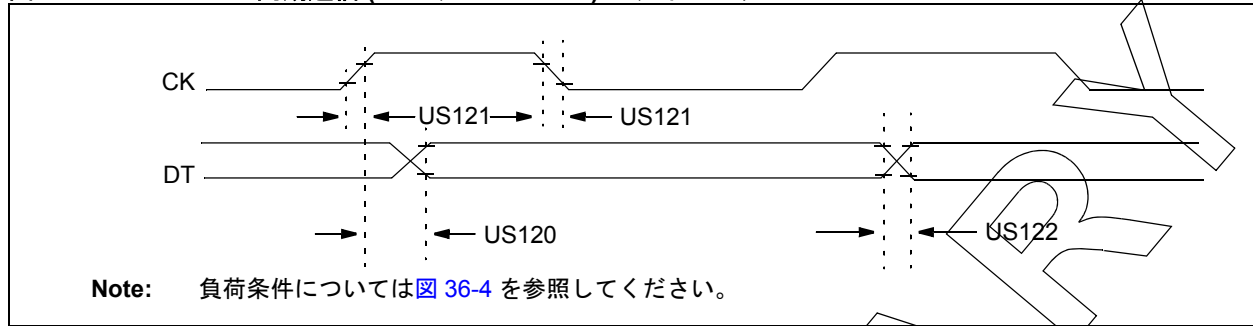


表 36-23: EUSART 同期送信の要件

標準動作条件 (特に明記しない場合)						
パラメータ No.	記号	特性	最小	最大	単位	条件
US120	TckH2DTV	同期送信 (マスタとスレーブ)	—	80	ns	$3.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
		クロック High からデータ出力確定までの時間	—	100	ns	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
US121	TckRF	クロック出力の立ち上がり/立ち下がり時間 (マスタモード)	—	45	ns	$3.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
			—	50	ns	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
US122	TdTRF	データ出力の立ち上がり/立ち下がり時間	—	45	ns	$3.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
			—	50	ns	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$

図 36-16: EUSART 同期受信 (マスタ/スレーブ) のタイミング

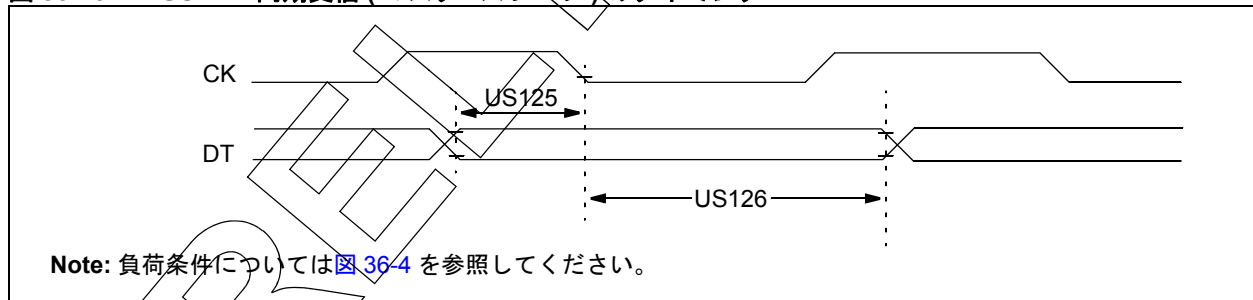


表 36-24: EUSART 同期受信の要件

標準動作条件 (特に明記しない場合)						
パラメータ No.	記号	特性	最小	最大	単位	条件
US125	TdTV2CKL	同期受信 (マスタ/スレーブ) CK ↓ 前のデータセットアップ (DT ホールド時間)	10	—	ns	
US126	TckL2DTL	CK ↓ 後のデータホールド (DT ホールド時間)	15	—	ns	

PIC16(L)F1764/5/8/9

図 36-17: SPI マスタモードのタイミング (CKE = 0、SMP = 0)

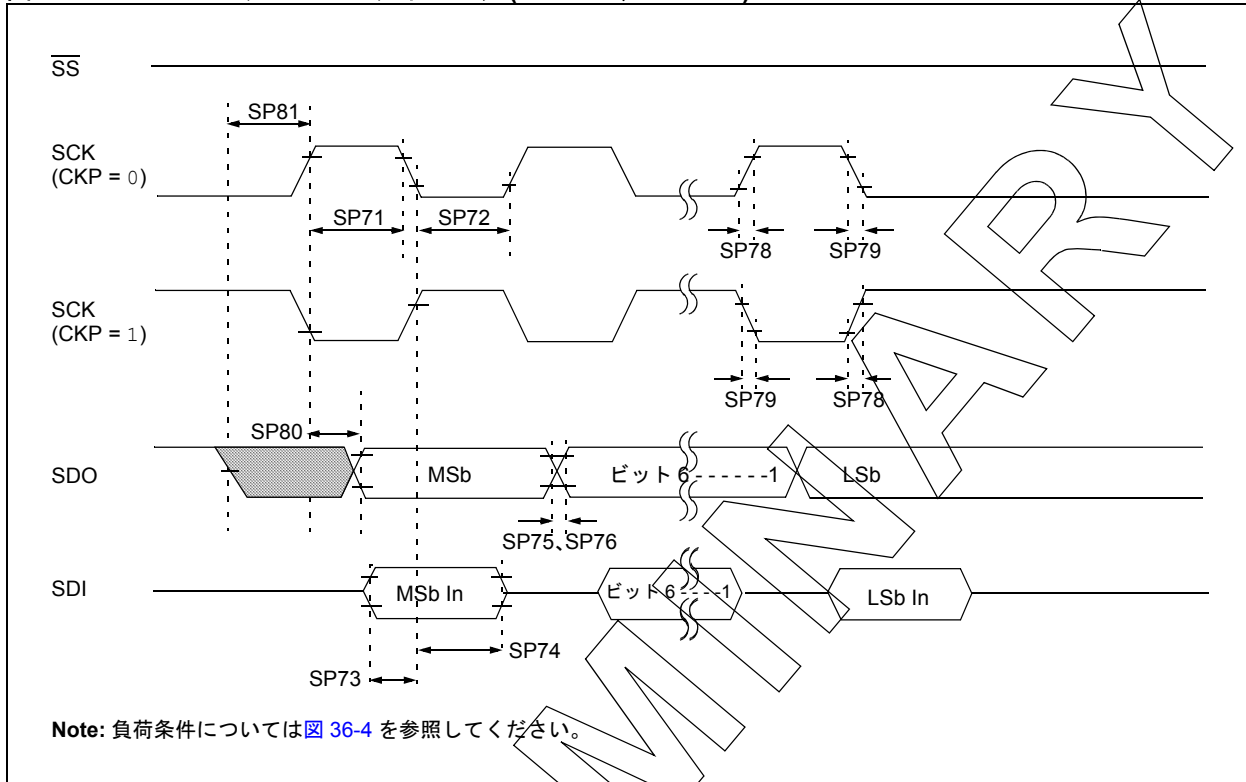


図 36-18: SPI マスタモードのタイミング (CKE = 1、SMP = 1)

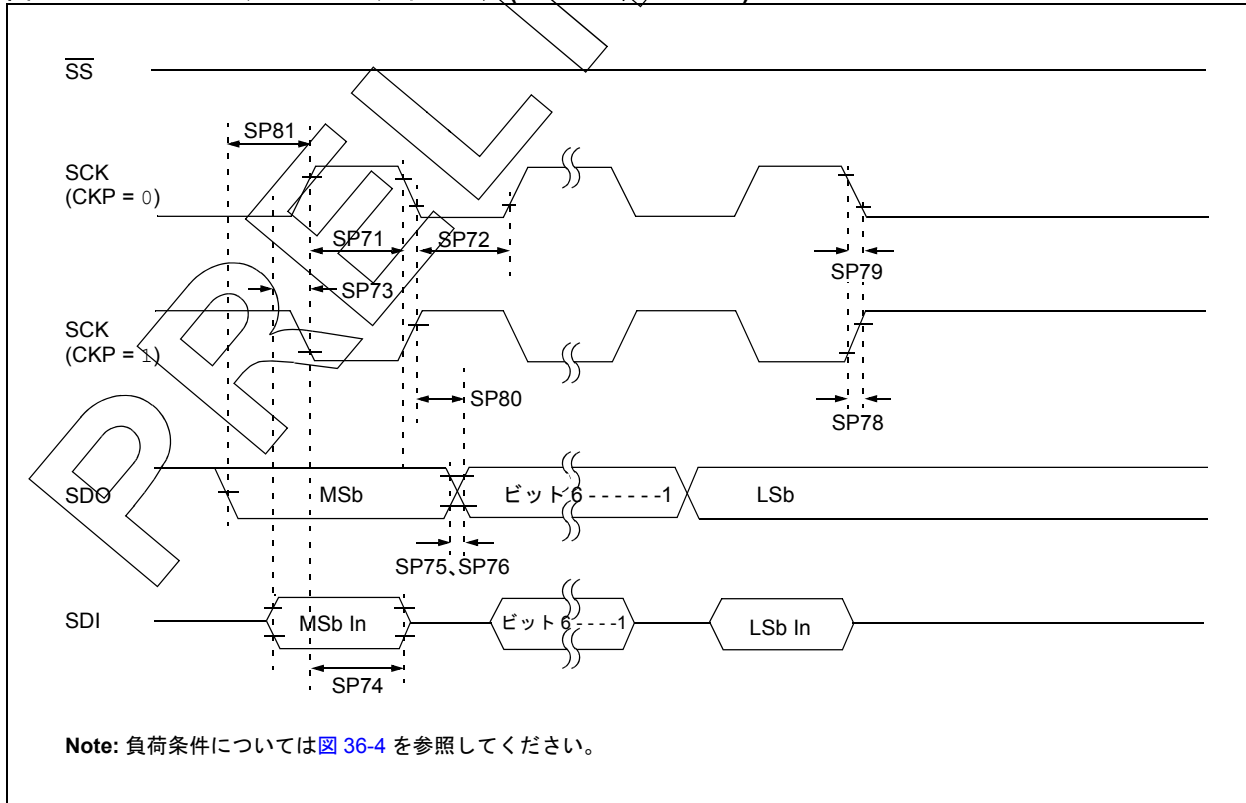


図 36-19: SPI スレーブモードのタイミング (CKE = 0)

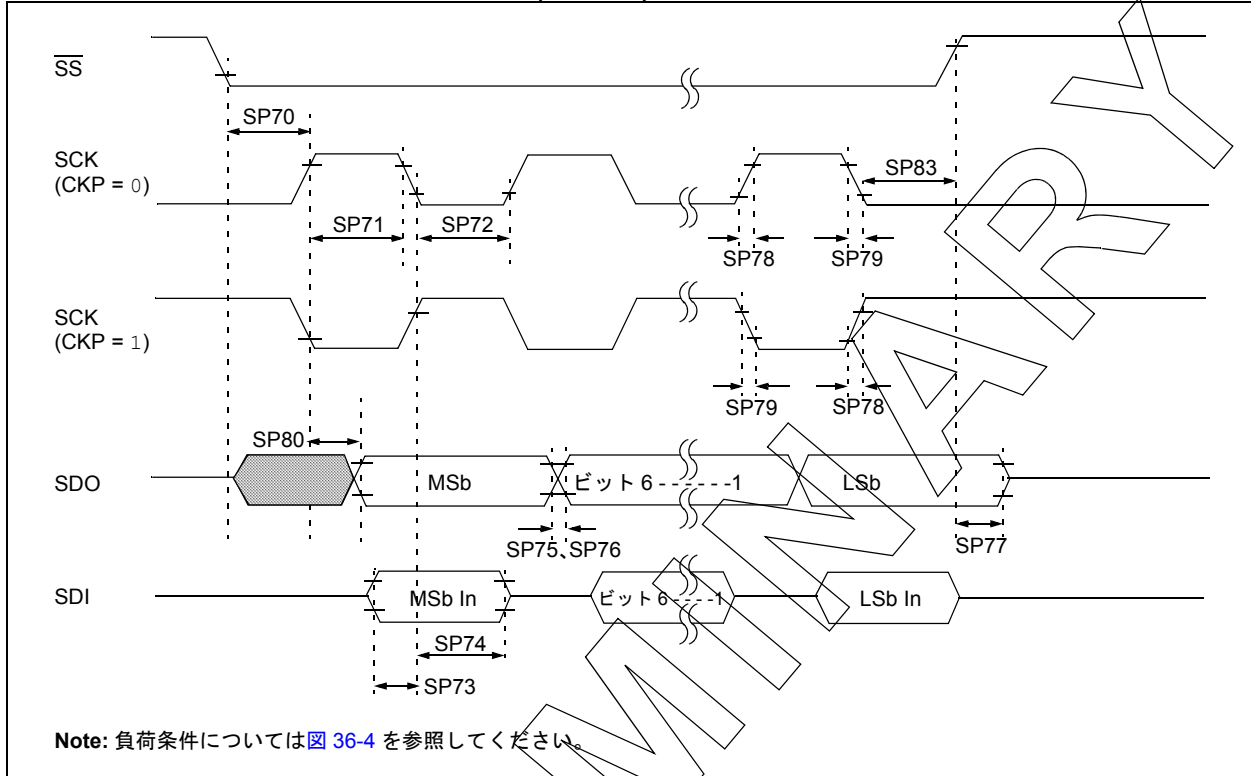
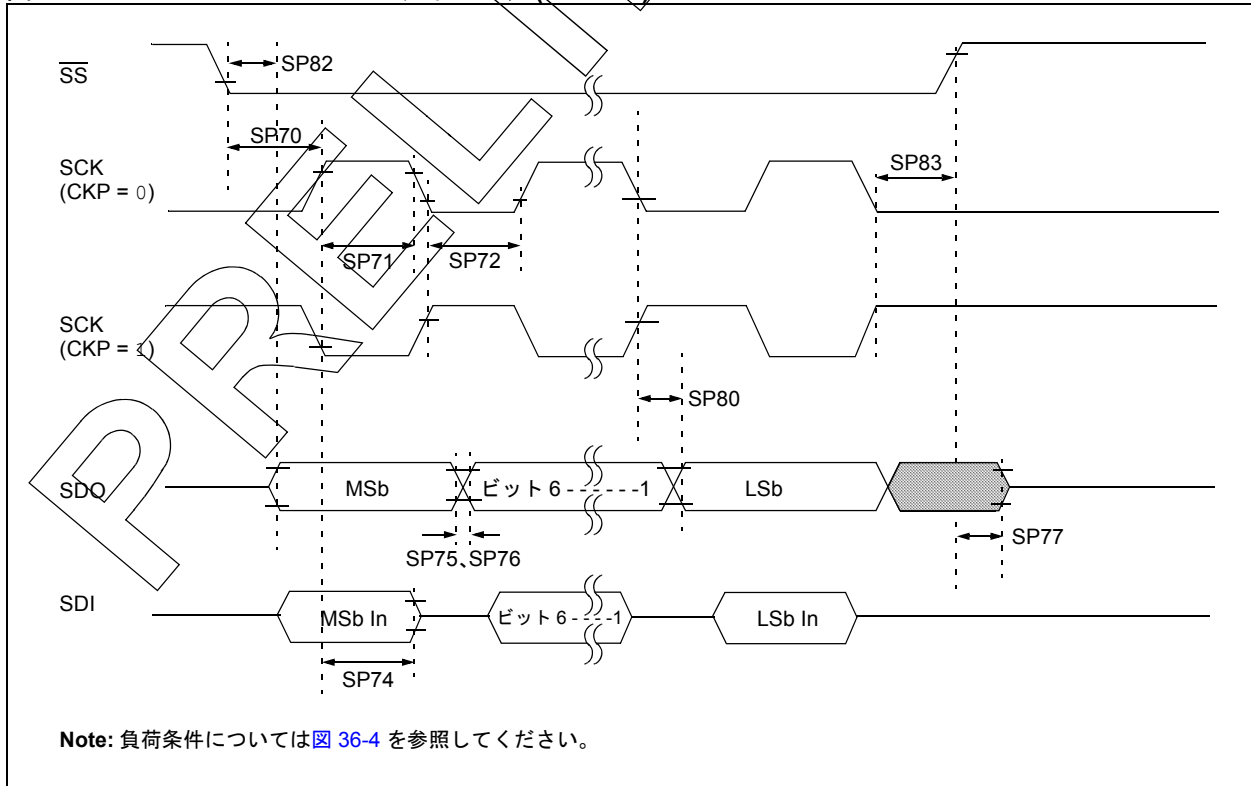


図 36-20: SPI スレーブモードのタイミング (CKE = 1)



PIC16(L)F1764/5/8/9

表 36-25: SPI モードの要件

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.†	最大	単位	条件
SP70*	TssL2sCH、TssL2sCL	$\overline{SS}\downarrow$ から SCK \downarrow または SCK 入力までの時間	Tcy	—	—	ns	
SP71*	TsCH	SCK 入力 High 時間 (スレーブモード)	Tcy + 20	—	—	ns	
SP72*	TsCL	SCK 入力 Low 時間 (スレーブモード)	Tcy + 20	—	—	ns	
SP73*	TdIV2sCH、TdIV2sCL	SDI データ入力から SCK エッジまでのセットアップ時間	100	—	—	ns	
SP74*	TsCH2dIL、TsCL2dIL	SDI データ入力から SCK エッジまでのホールド時間	100	—	—	ns	
SP75*	TdoR	SDO データ出力の立ち上がり時間	—	10	25	ns	3.0 V ≤ VDD ≤ 5.5 V
			—	25	50	ns	1.8 V ≤ VDD ≤ 5.5 V
SP76*	TdoF	SDO データ出力の立ち下がり時間	—	10	25	ns	
SP77*	TssH2doZ	\overline{SS} から SDO 出力ハイインピーダンスまでの時間	10	—	50	ns	
SP78*	TscR	SCK 出力の立ち上がり時間 (マスターモード)	—	10	25	ns	3.0 V ≤ VDD ≤ 5.5 V
			—	25	50	ns	1.8 V ≤ VDD ≤ 5.5 V
SP79*	TscF	SCK 出力の立ち下がり時間 (マスターモード)	—	10	25	ns	
SP80*	TsCH2doV、TsCL2doV	SCK エッジから確定した SDO データ出力開始までの時間	—	—	50	ns	3.0 V ≤ VDD ≤ 5.5 V
			—	—	145	ns	1.8 V ≤ VDD ≤ 5.5 V
SP81*	TdoV2sCH、TdoV2sCL	SCK エッジ前の SDO データ出力セットアップ時間	1 Tcy	—	—	ns	
SP82*	TssL2doV	$\overline{SS}\downarrow$ エッジから確定した SDO データ出力開始までの時間	—	—	50	ns	
SP83*	TsCH2ssH、TsCL2ssH	SCK エッジから \overline{SS} までの時間	1.5 Tcy + 40	—	—	ns	

* これらのパラメータは特性データであり、テストしていません。

† 特に明記しない限り、「Typ.」列のデータは 3.0 V/25 °C での値です。これらのパラメータは設計上の目安であり、テストしていません。

図 36-21: I²C™ バスのスタート/ストップビットのタイミング

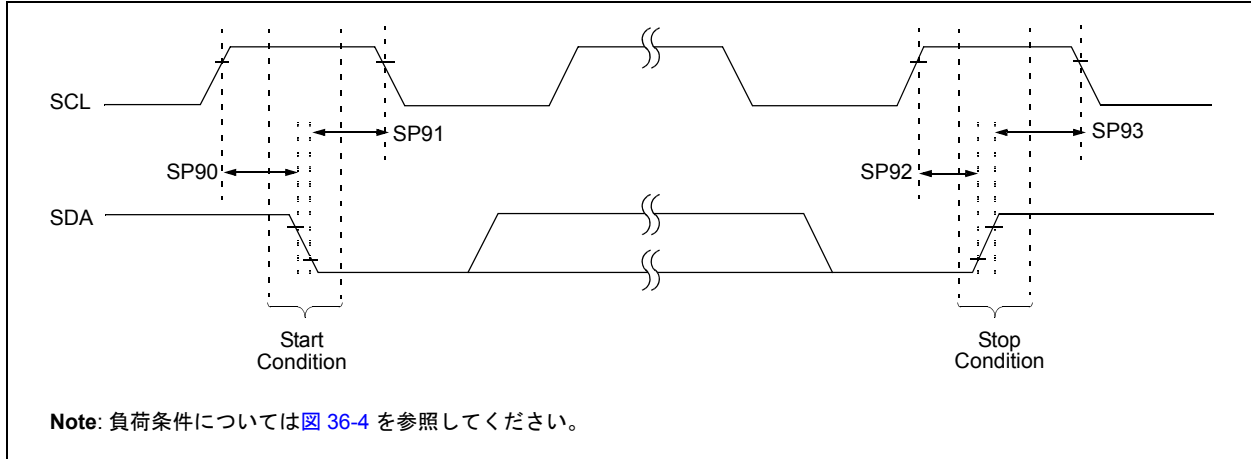


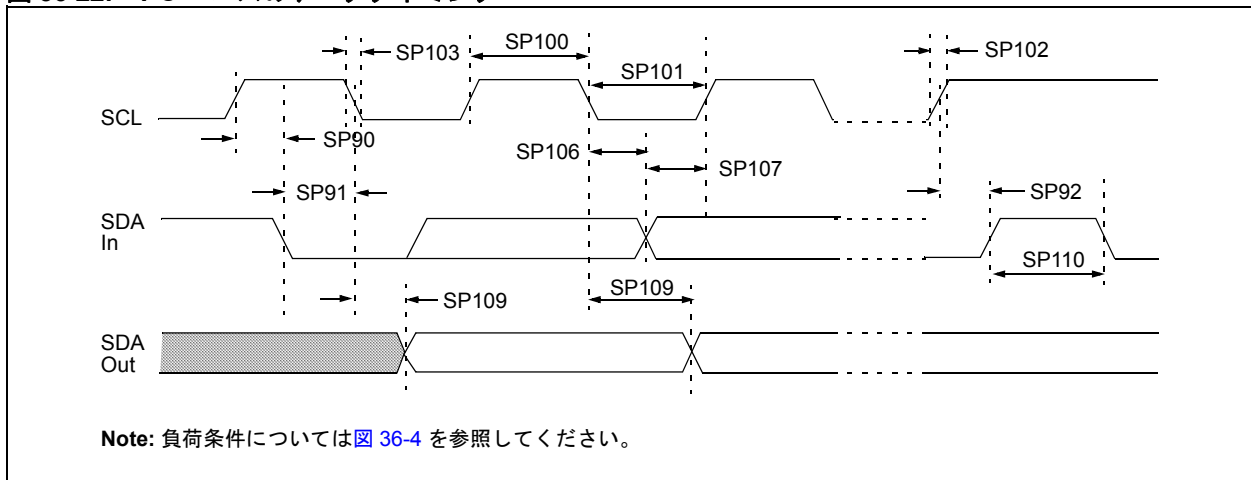
表 36-26: I²C™ バスのスタート/ストップビットの要件

標準動作条件 (特に明記しない場合)

パラメータ No.	記号	特性	最小	Typ.	最大	単位	条件	
SP90*	TSU:STA	スタート条件	100 kHz モード	4700	—	—	ns	反復スタート条件の場合のみ該当
		セットアップ時間	400 kHz モード	600	—	—		
SP91*	THD:STA	スタート条件	100 kHz モード	4000	—	—	ns	この期間後、最初のクロックパルスが生成される
		ホールド時間	400 kHz モード	600	—	—		
SP92*	TSU:STO	ストップ条件	100 kHz モード	4700	—	—	ns	
		セットアップ時間	400 kHz モード	600	—	—		
SP93	THD:STO	ストップ条件	100 kHz モード	4000	—	—	ns	
		ホールド時間	400 kHz モード	600	—	—		

* これらのパラメータは特性データであり、テストしていません。

図 36-22: I²C™ バスのデータタイミング



PIC16(L)F1764/5/8/9

表 36-27: I²C™ バスデータの要件

標準動作条件 (特に明記しない場合)							
パラメータ No.	記号	特性		最小	最大	単位	条件
SP100*	THIGH	クロック High 時間	100 kHz モード	4.0	—	μs	デバイスの動作周波数が 1.5 MHz 以上である事が必要
			400 kHz モード	0.6	—	μs	デバイスの動作周波数が 10 MHz 以上である事が必要
			SSP モジュール	1.5 Tcy	—		
SP101*	TLOW	クロック Low 時間	100 kHz モード	4.7	—	μs	デバイスの動作周波数が 1.5 MHz 以上である事が必要
			400 kHz モード	1.3	—	μs	デバイスの動作周波数が 10 MHz 以上である事が必要
			SSP モジュール	1.5 Tcy	—		
SP102*	TR	SDA および SCL 立ち上がり時間	100 kHz モード	—	1000	ns	
			400 kHz モード	20 + 0.1 C _B	300	ns	C _B の仕様は 10 ~ 400 pF
SP103*	TF	SDA および SCL 立ち下がり時間	100 kHz モード	—	250	ns	
			400 kHz モード	20 + 0.1 C _B	250	ns	C _B の仕様は 10 ~ 400 pF
SP106*	THD:DAT	データ入力 ホールド時間	100 kHz モード	0	—	ns	
			400 kHz モード	0	0.9	μs	
SP107*	TSU:DAT	データ入力セットアップ時間	100 kHz モード	250	—	ns	(Note 2)
			400 kHz モード	100	—	ns	
SP109*	TAA	クロックから出力 確定	100 kHz モード	—	3500	ns	(Note 1)
			400 kHz モード	—	—	ns	
SP110*	TBUF	バスフリー時間	100 kHz モード	4.7	—	μs	新しい送信を開始する前にバスが解放されていない時間
			400 kHz モード	1.3	—	μs	
SP111	CB	バス容量性負荷		—	400	pF	

* これらのパラメータは特性データであり、テストしていません。

Note 1: デバイスがトランスマッタの場合、意図しないスタート/ストップ条件の発生を防ぐために SCL の立ち下がりエッジの未定義領域をブリッジする内部最小遅延時間 (min. 300 ns) を確保する必要があります。

2: 高速モード (400 kHz) の I²C™ バスデバイスは標準モード (100 kHz) の I²C バスシステムでも使えますが、TSU:DAT ≥ 250 ns の要件を満たす必要があります。デバイスが SCL 信号の Low 周期をストレッチしない場合、この要件を自動的に満たします。SCL 信号の Low 周期をストレッチする場合、標準モードの I²C バス仕様に従い、SCL ラインを解放する TR max. + TSU:DAT = 1000 + 250 = 1250 ns 前に次のデータビットを SDA ラインへ出力する必要があります。

37.0 DC および AC 特性の図表

ここに示す図表は設計上の目安であり、テストしていません。

下図表の一部は、仕様動作レンジ外で計測されたデータを含みます (例: 仕様レンジ外の VDD を使用)。これらのデータは情報提供のみを目的として掲載するものであり、デバイスの適切な動作が保証されるのは仕様レンジ内のみです。

特に明記しない限り、図は全て F デバイスと LF デバイスの両方に対応しています。

Note: 以下の図表は限られたサンプル数に基づく統計的な結果であり、あくまでも情報提供を目的としています。ここに記載した性能特性は検査されておらず保証されません。下図表の一部には、仕様動作レンジ外で計測されたデータも含みます (例: 仕様レンジ外の電源を使用)。従ってこれらのデータは保証範囲外です。

「Typ.」は 25 °C における分布の平均値を表します。「最大値」と「最小値」は、それぞれ (平均値 + 3 σ) と (平均値 - 3 σ) を表します。 σ は各仕様温度レンジを通じての標準偏差です。

PIC16(L)F1764/5/8/9

図 37-1: 各温度における V_{OH} と I_{OH} 、 $V_{DD} = 5.5\text{ V}$ 、PIC16F1764/5/8/9 のみ

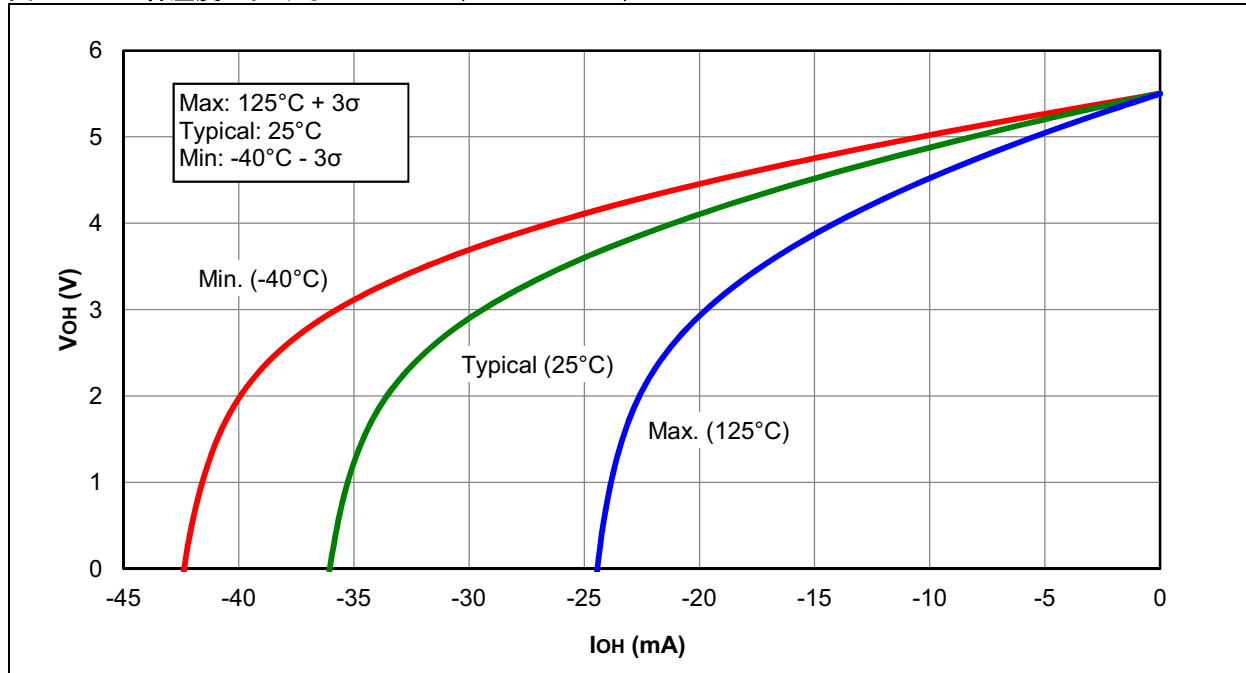


図 37-2: 各温度における V_{OL} と I_{OL} 、 $V_{DD} = 5.5\text{ V}$ 、PIC16F1764/5/8/9 のみ

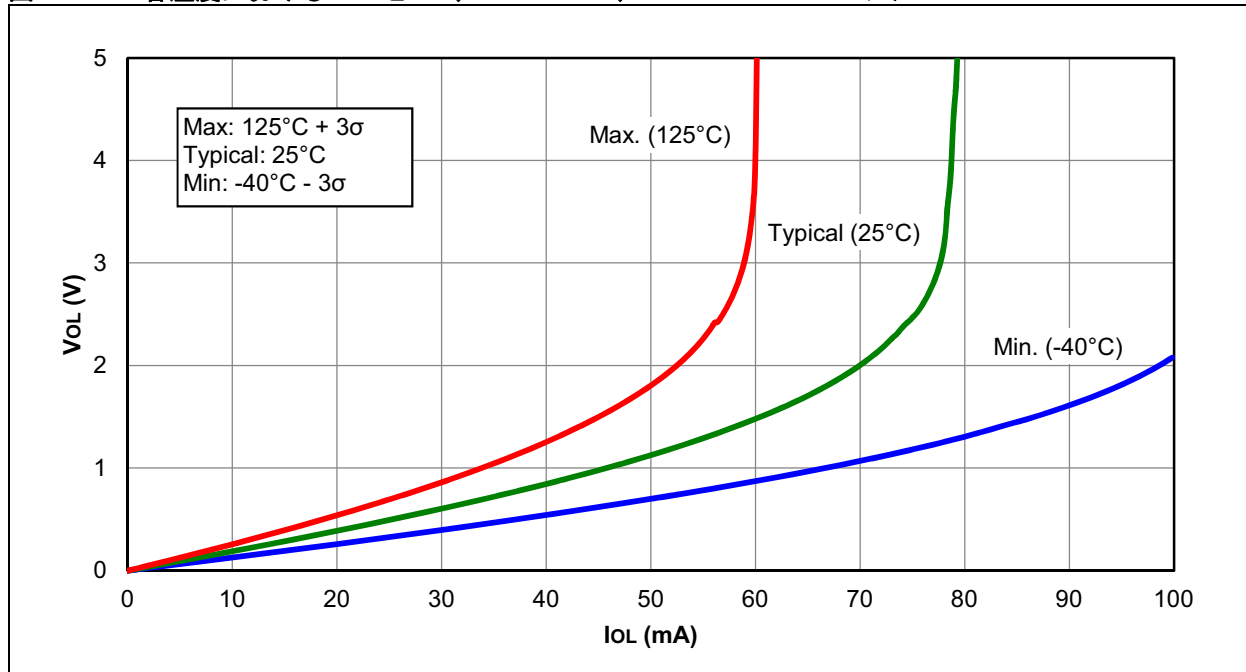


図 37-3: 各温度における V_{OH} と I_{OH} 、 $V_{DD} = 3.0\text{ V}$

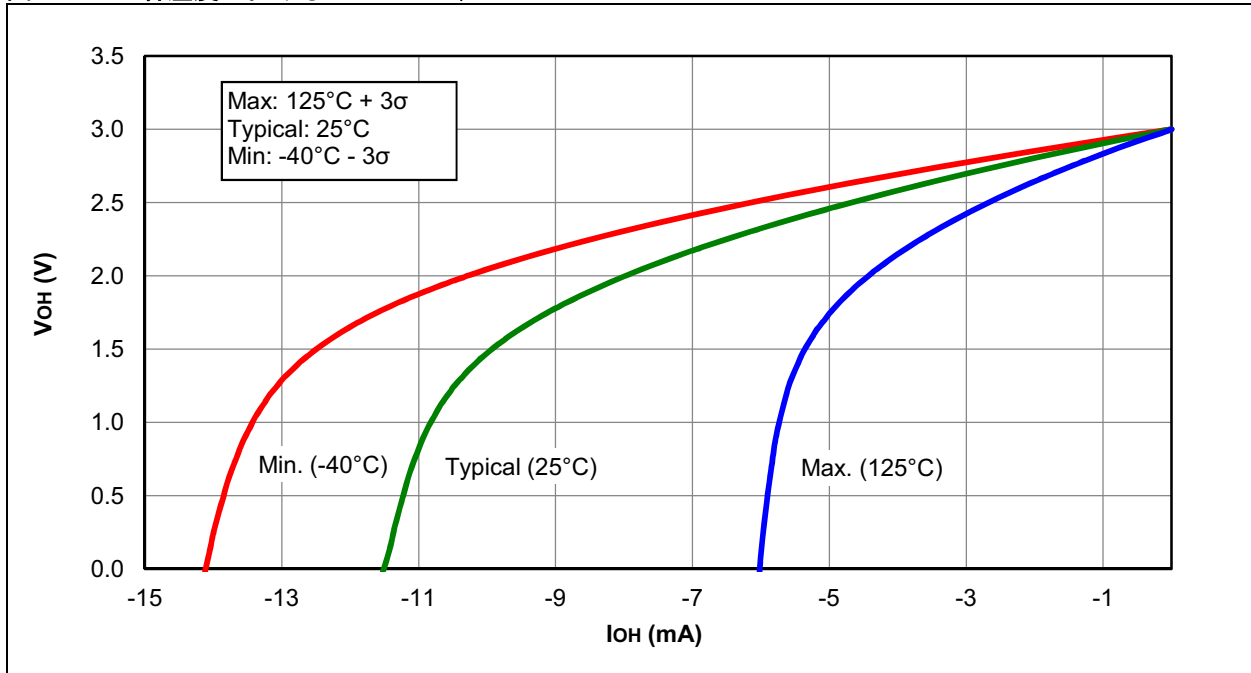
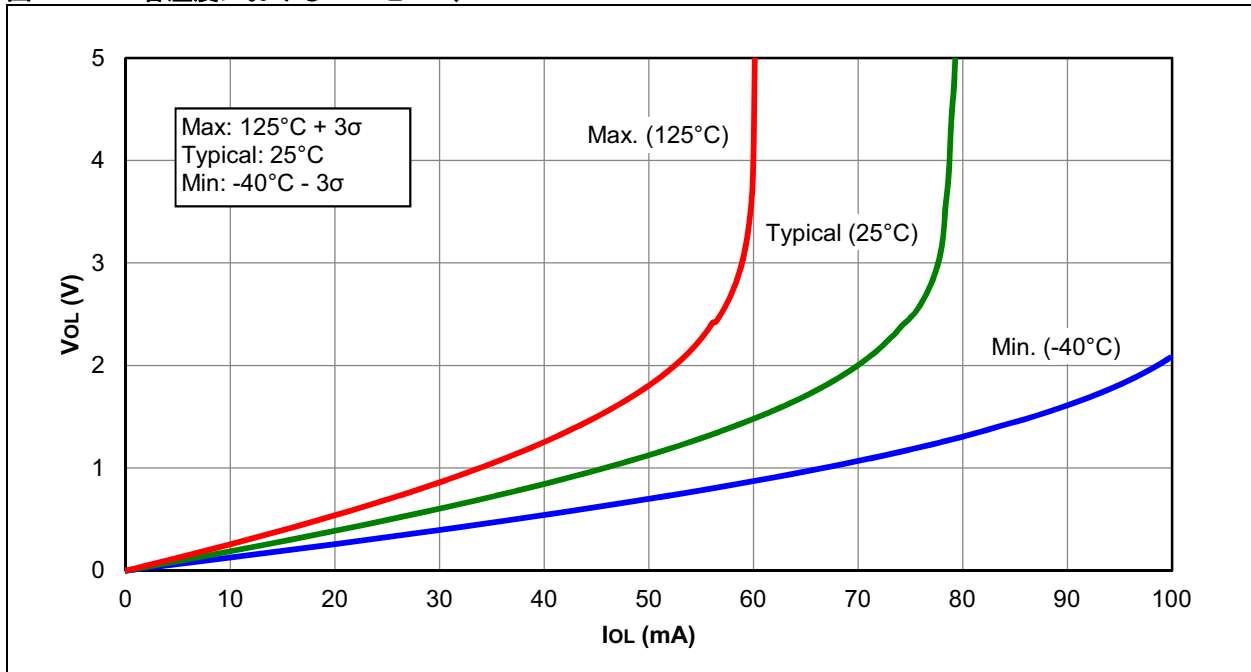


図 37-4: 各温度における V_{OL} と I_{OL} 、 $V_{DD} = 3.0\text{ V}$



PIC16(L)F1764/5/8/9

図 37-5: 各温度における V_{OH} と I_{OH} 、 $V_{DD} = 1.8V$ 、PIC16LF1764/5/8/9 のみ

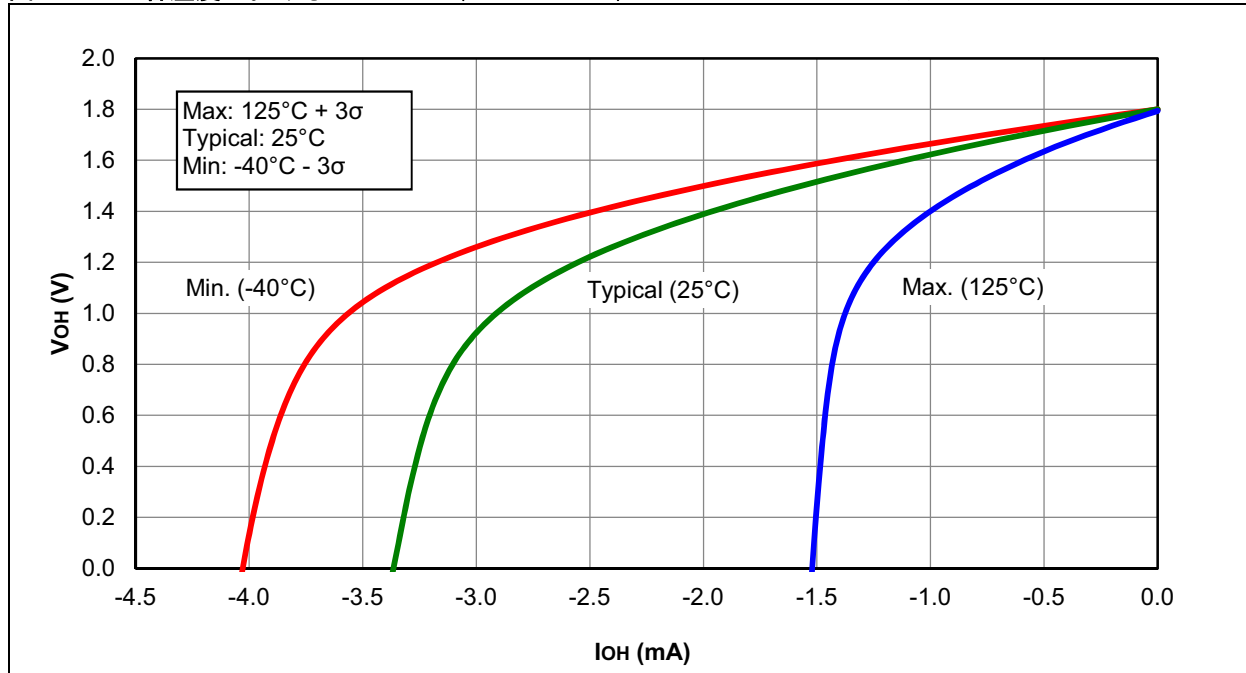


図 37-6: 各温度における V_{OL} と I_{OL} 、 $V_{DD} = 1.8V$ 、PIC16LF1764/5/8/9 のみ

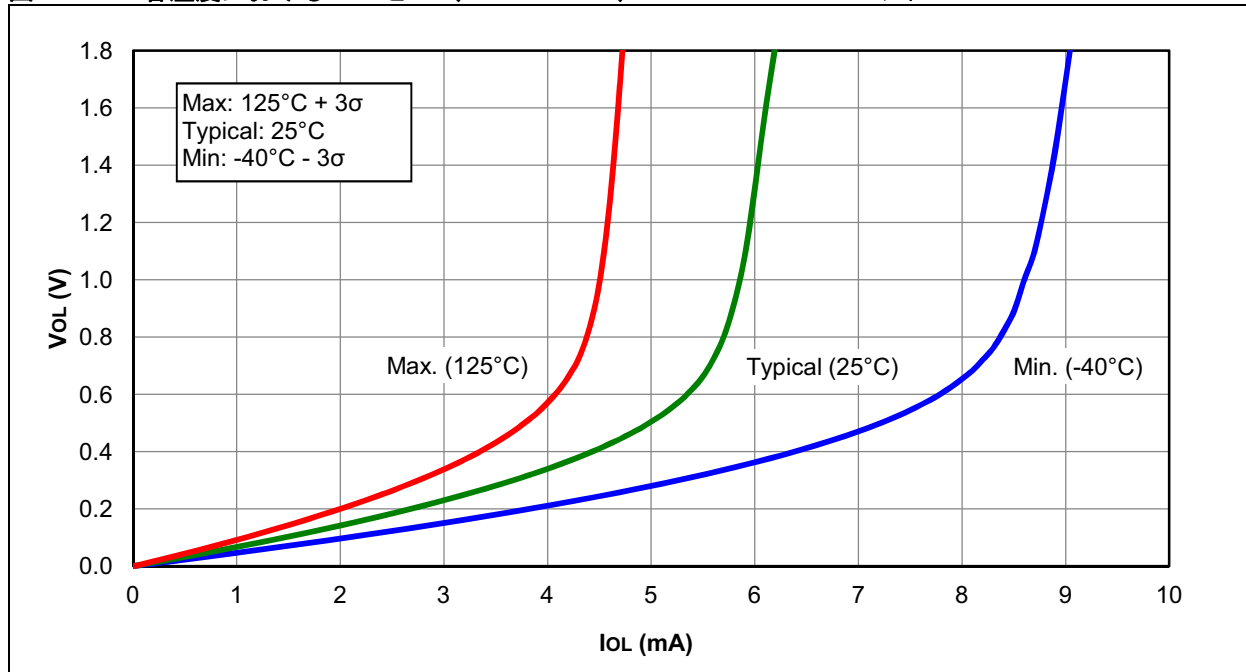


図 37-7: POR リリース電圧

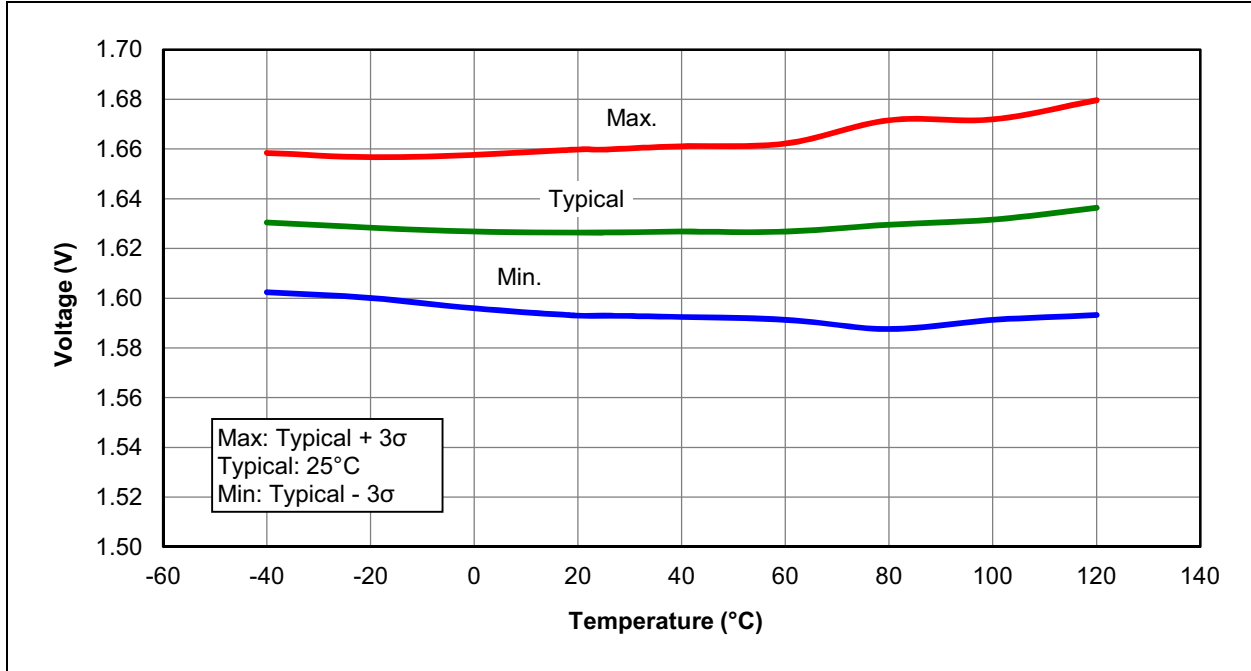
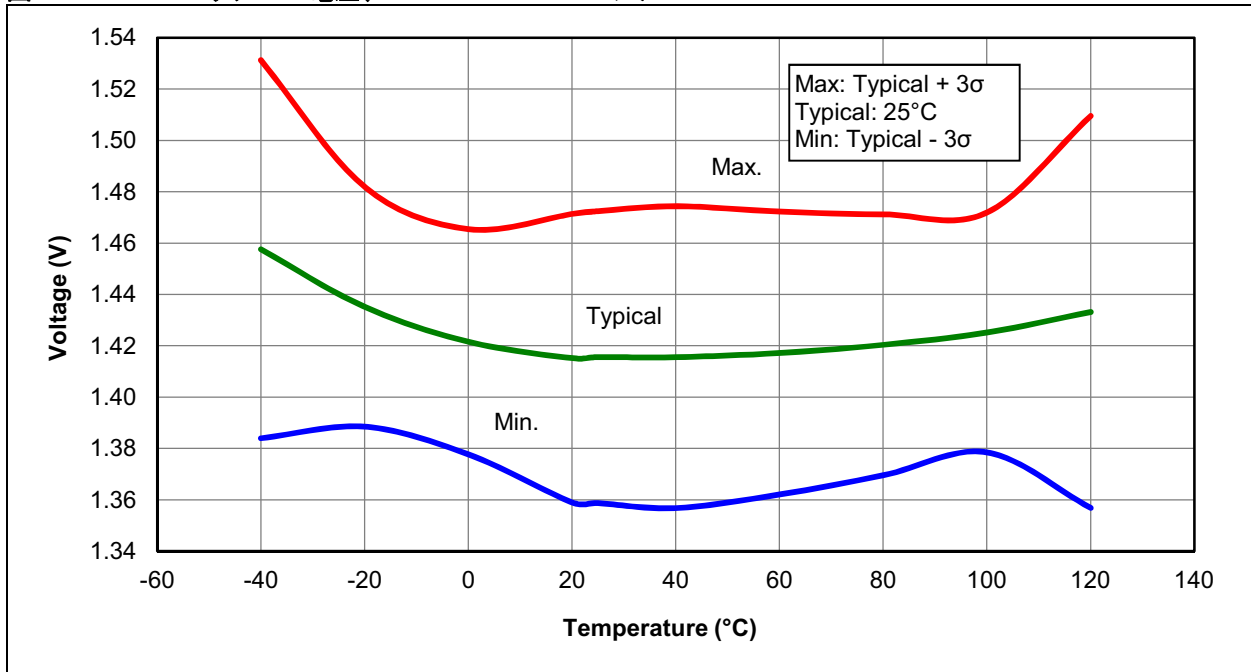


図 37-8: POR リアーム電圧、PIC16F1764/5/8/9 のみ



PIC16(L)F1764/5/8/9

図 37-9: ブラウンアウトリセット電圧、BORV = 1、PIC16LF1764/5/8/9 のみ

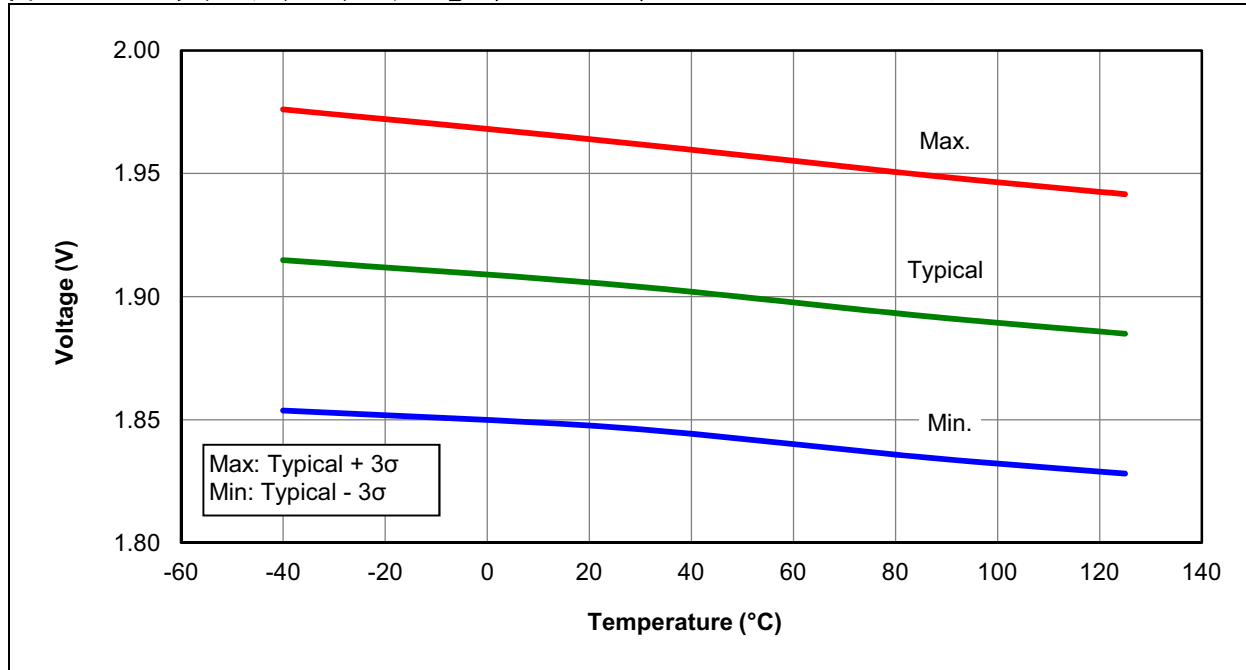


図 37-10: ブラウンアウトリセットヒステリシス、BORV = 1、PIC16LF1764/5/8/9 のみ

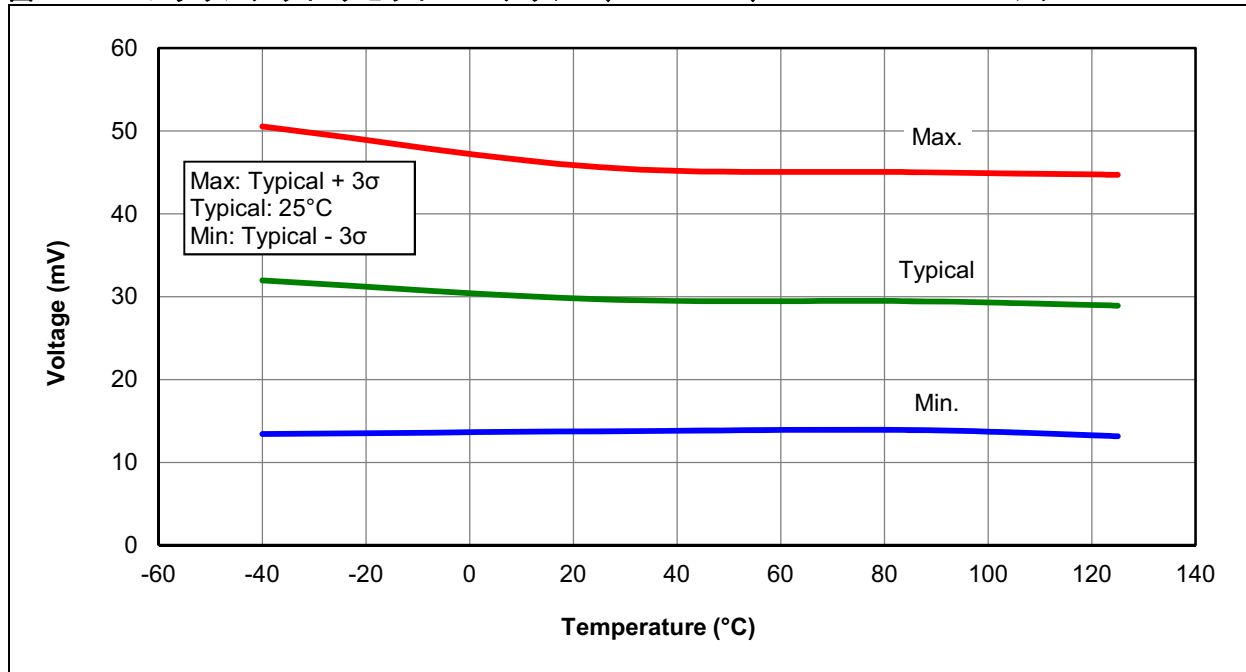


図 37-11: ブラウンアウトリセット電圧、BORV = 1、PIC16F1764/5/8/9 のみ

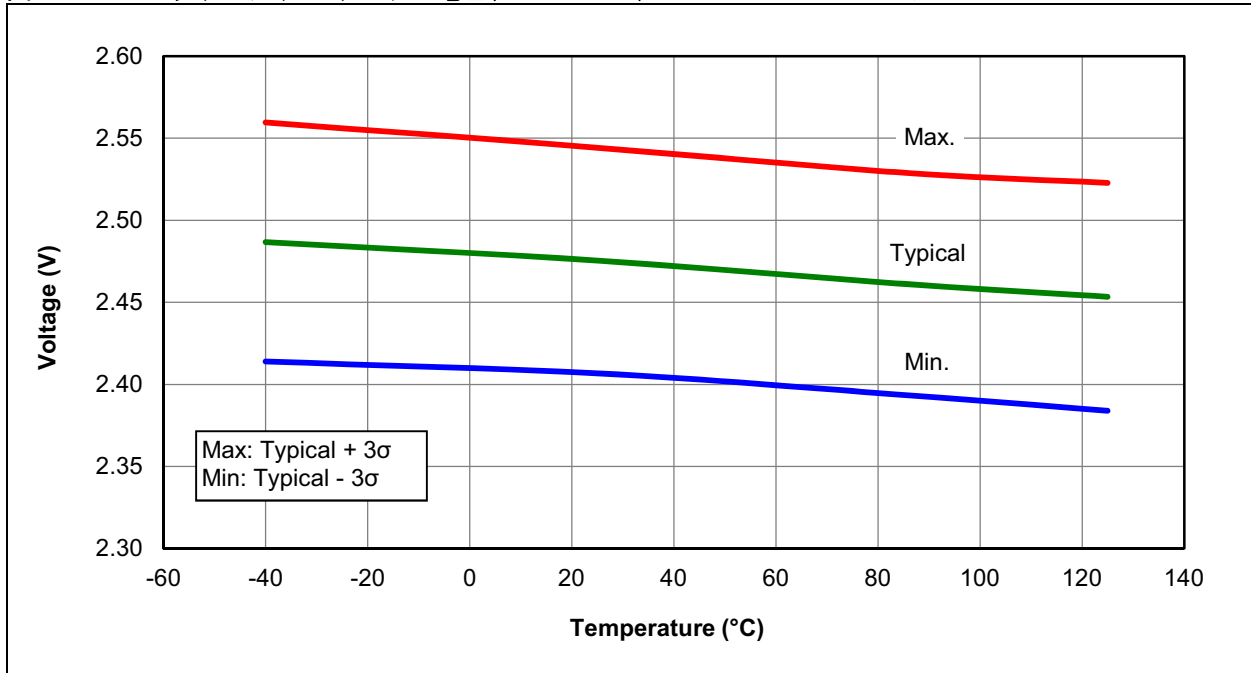
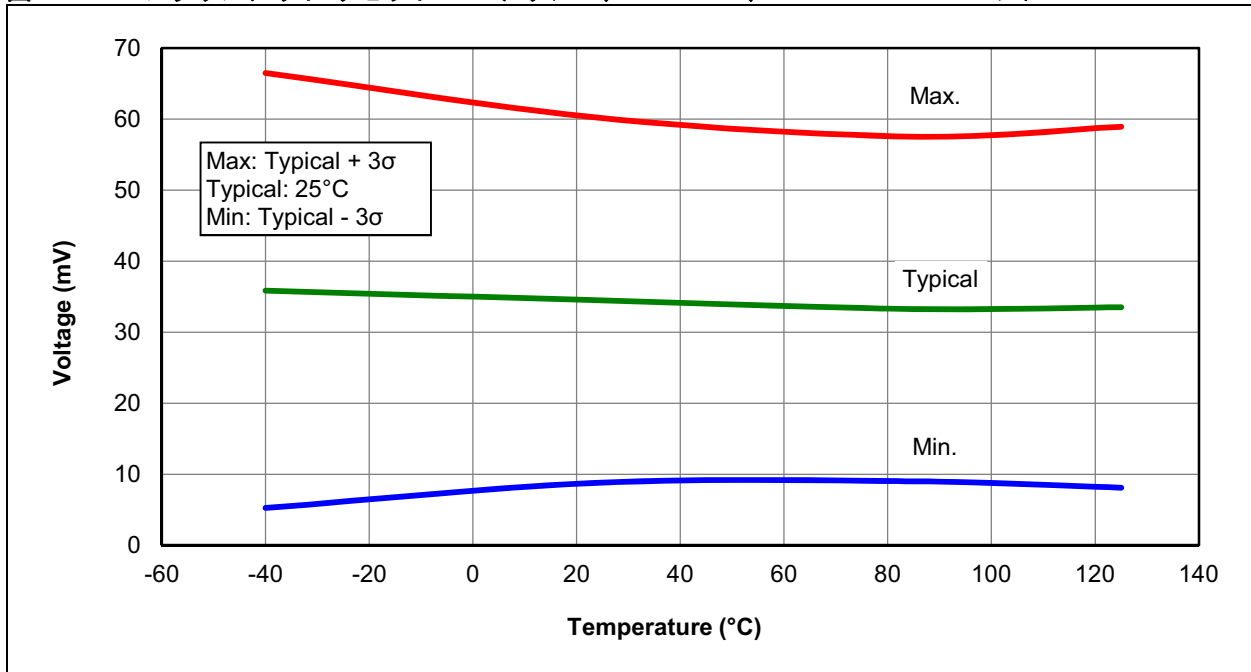


図 37-12: ブラウンアウトリセットヒステリシス、BORV = 1、PIC16F1764/5/8/9 のみ



PIC16(L)F1764/5/8/9

図 37-13: ブラウンアウト リセット電圧、BORV = 0

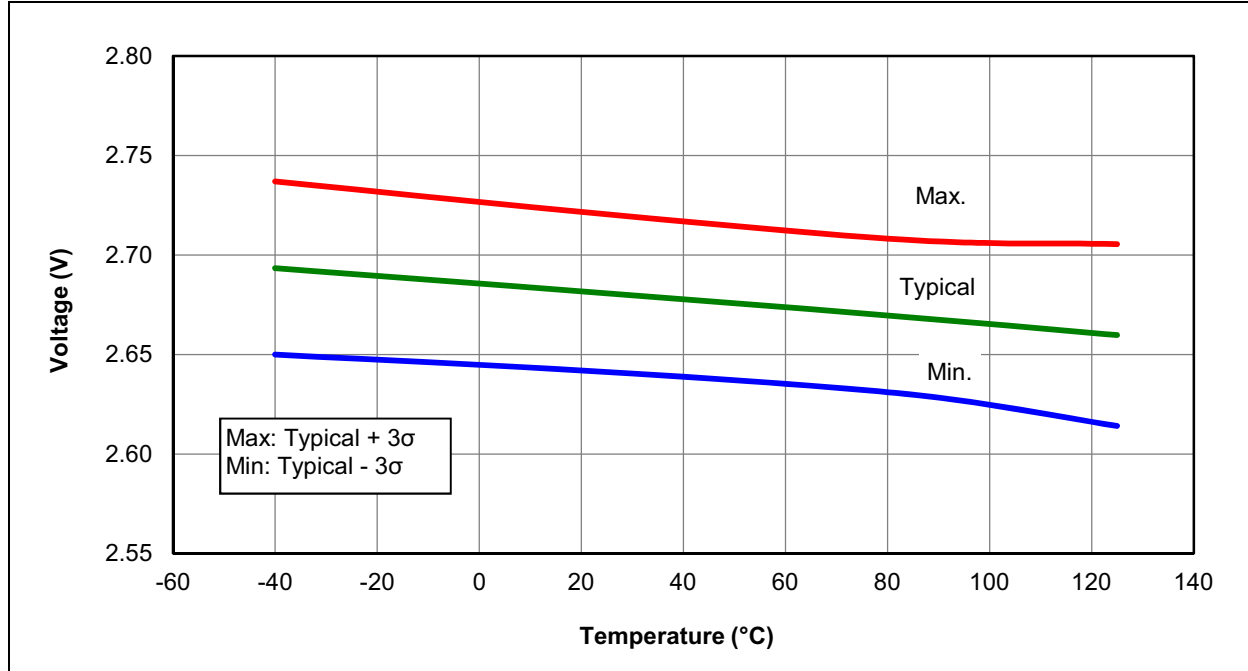


図 37-14: ブラウンアウト リセット ヒステリシス、BORV = 0

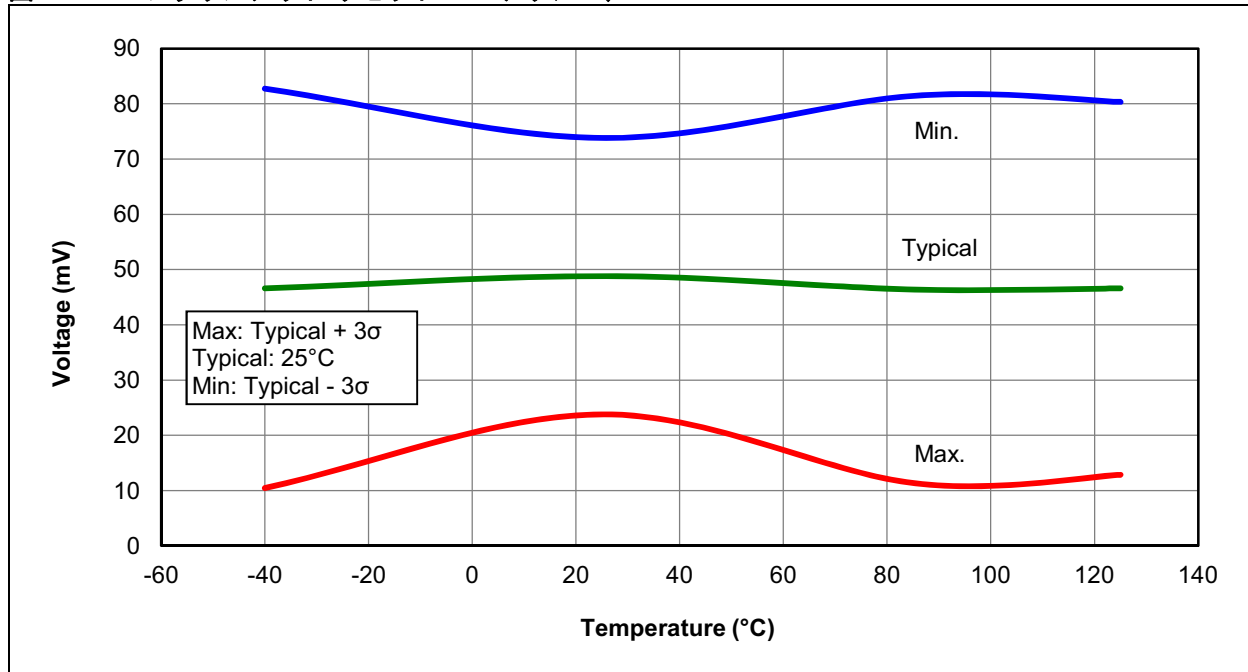


図 37-15: 低消費電力ブラウンアウトリセット電圧、LPBOR = 0

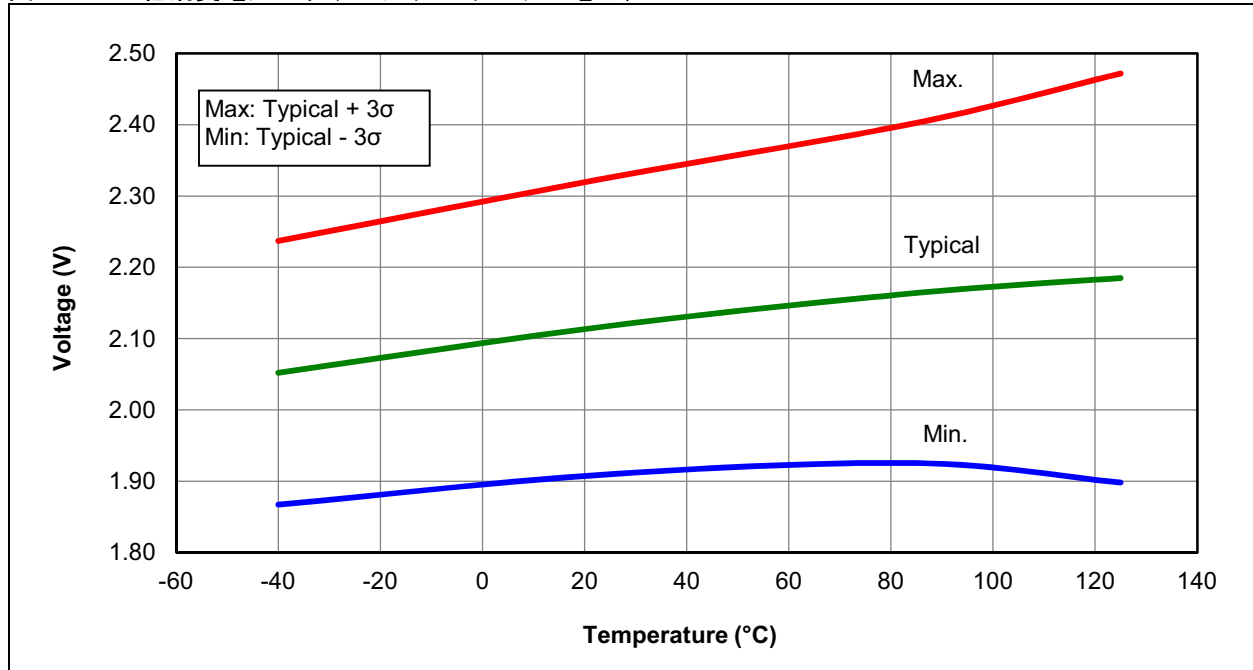
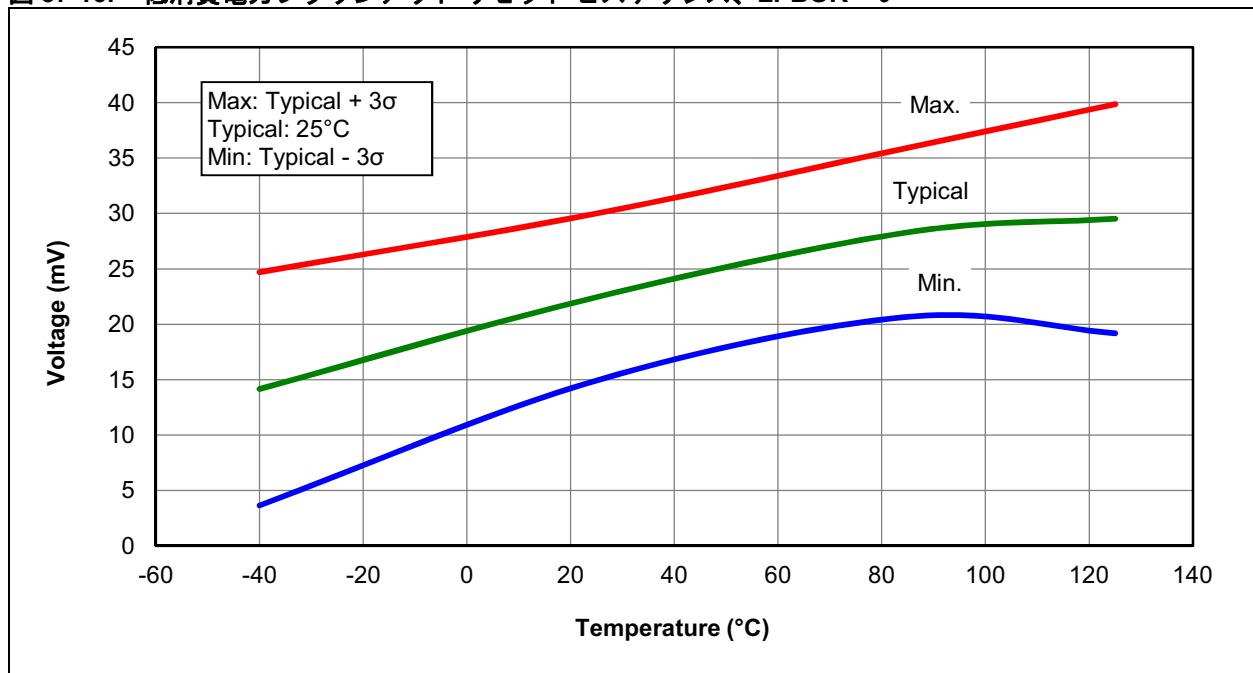


図 37-16: 低消費電力ブラウンアウトリセットヒステリシス、LPBOR = 0



PIC16(L)F1764/5/8/9

図 37-17: WDT タイムアウト時間

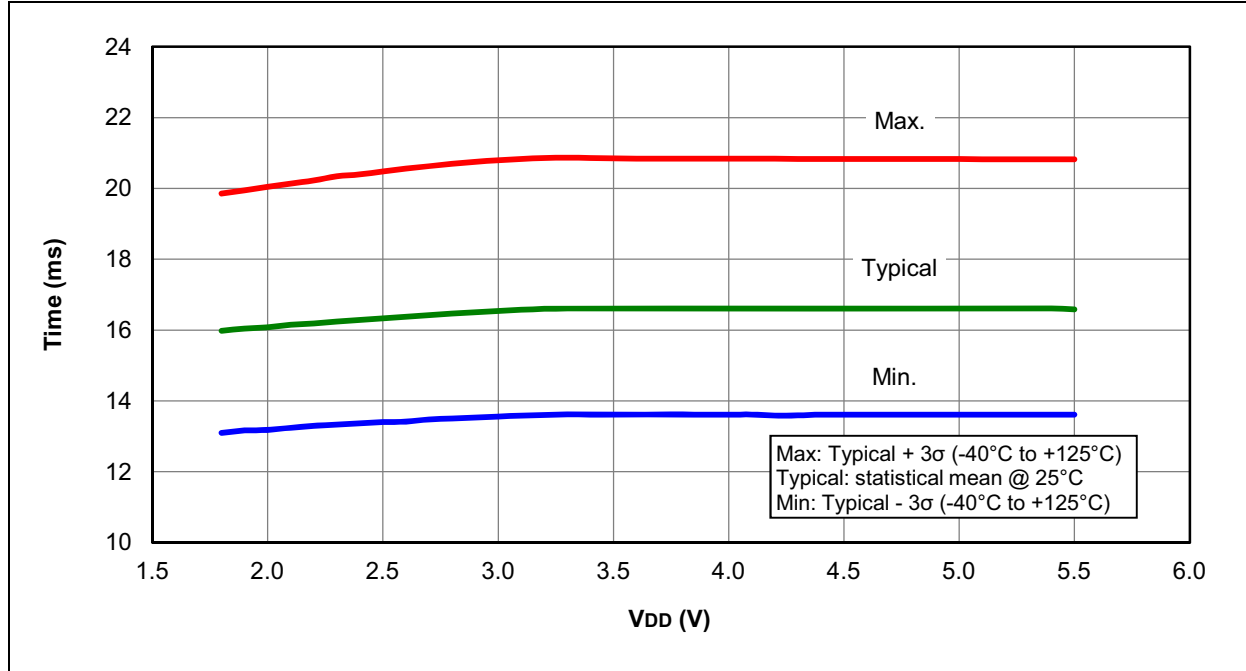


図 37-18: PWRT 時間

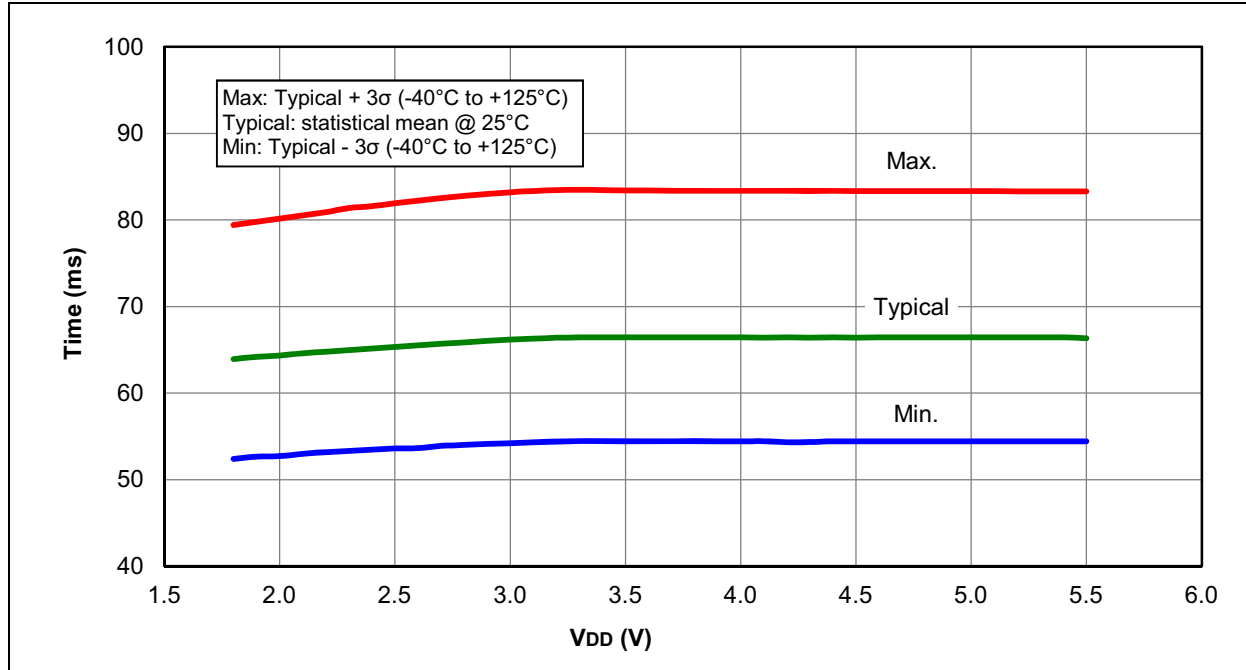


図 37-19: FVR 安定化時間

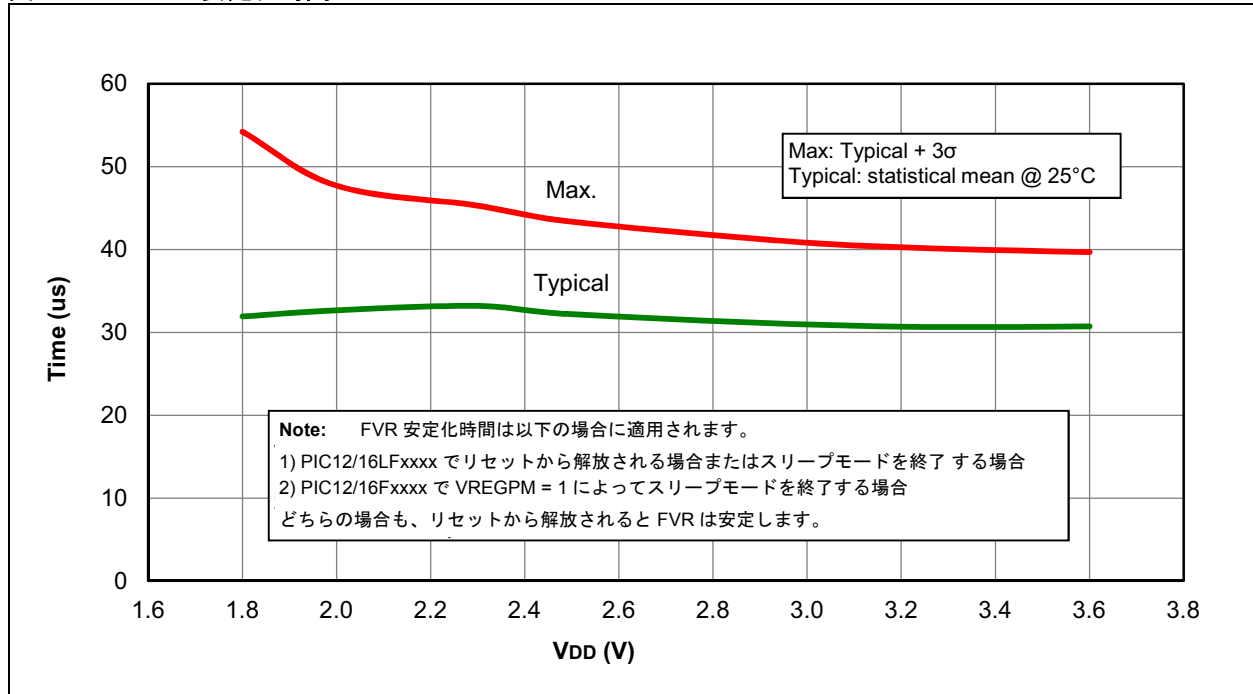
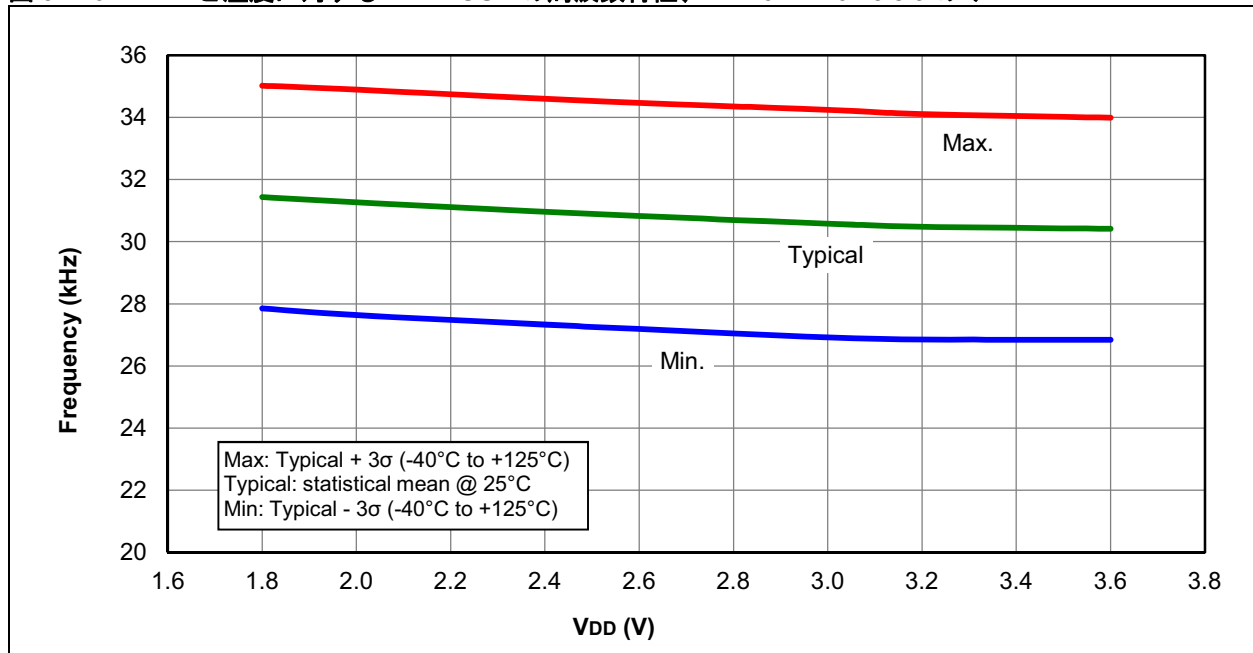


図 37-20: VDD と温度に対する LFINTOSC の周波数特性、PIC16LF1764/5/8/9 のみ



PIC16(L)F1764/5/8/9

図 37-21: VDD と温度に対する LFINTOSC の周波数特性、PIC16F1764/5/8/9 のみ

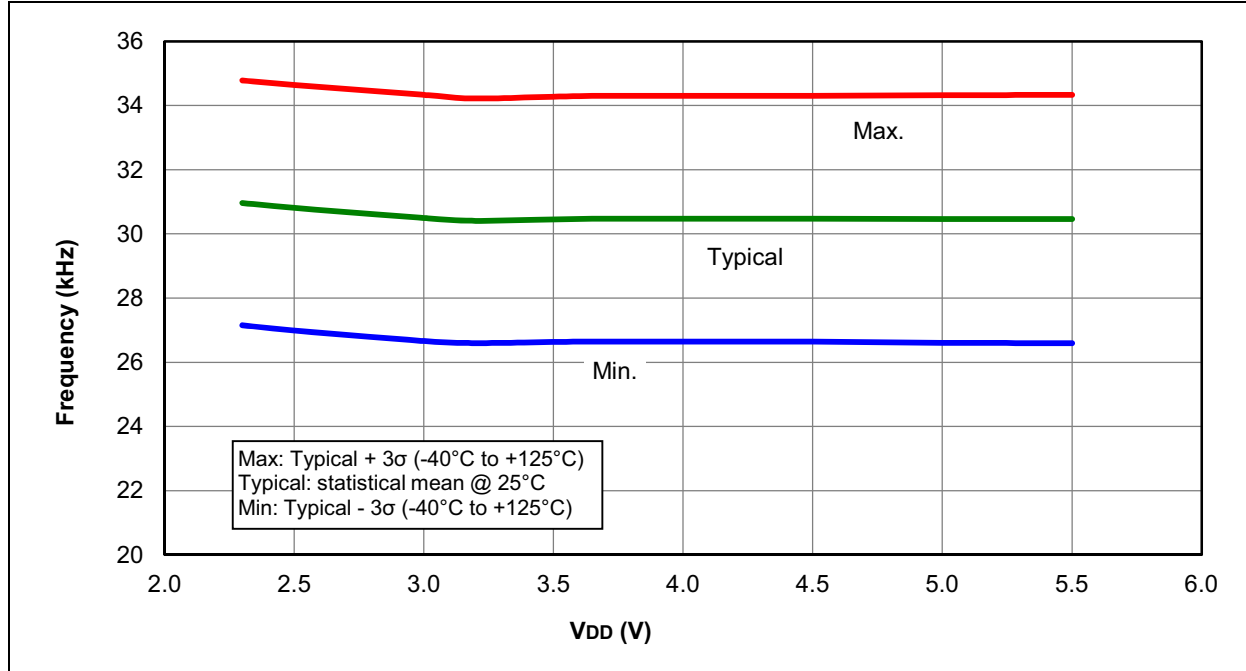


図 37-22: スリープ、HFINTOSC 信号源による復帰期間、PIC16LF1764/5/8/9 のみ

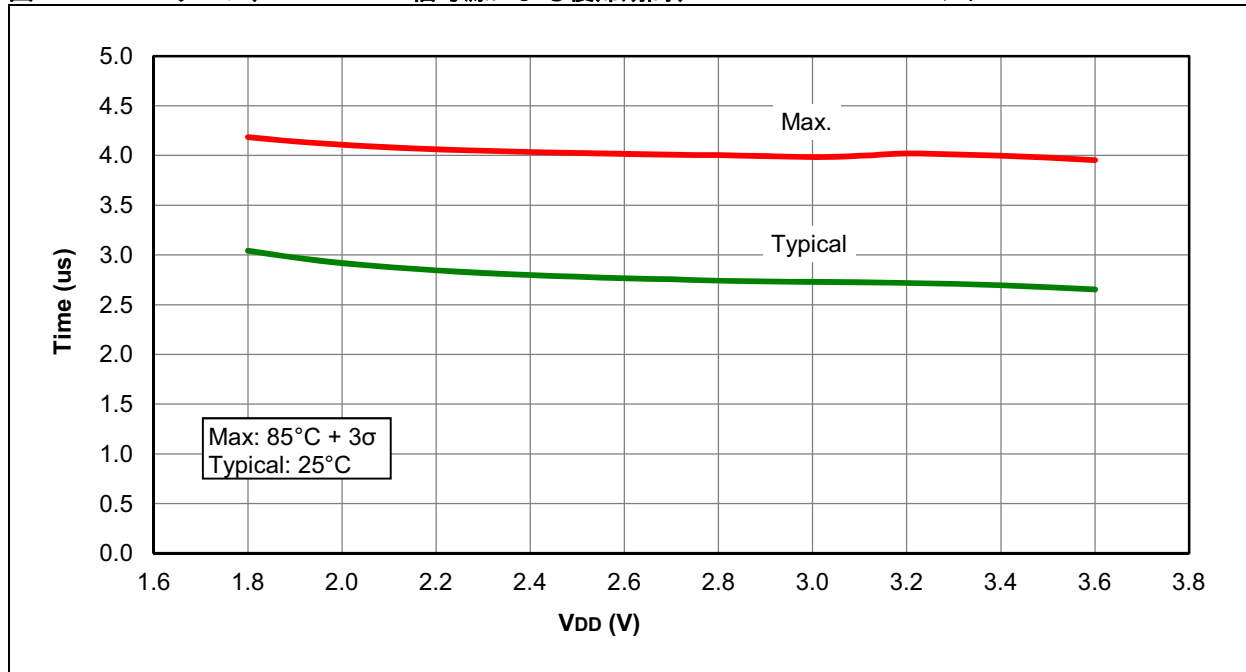


図 37-23: 低消費電力スリープ、HFINTOSC 信号源による復帰期間、VREGPM = 1、PIC16F1764/5/8/9 のみ

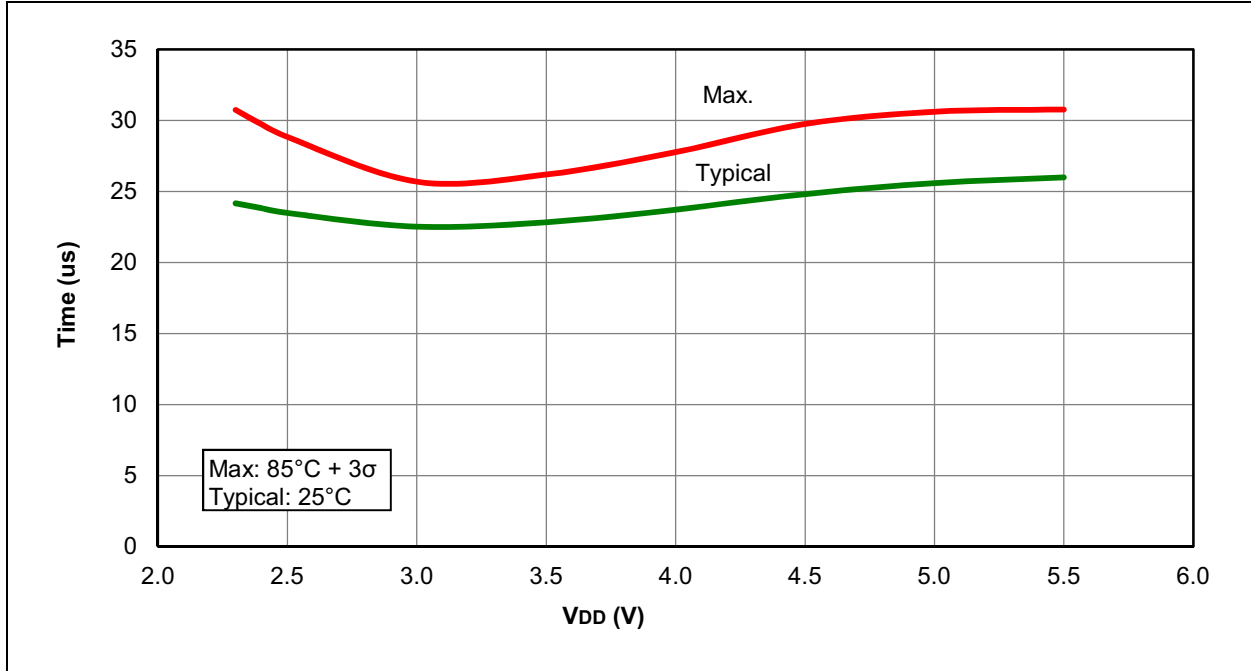
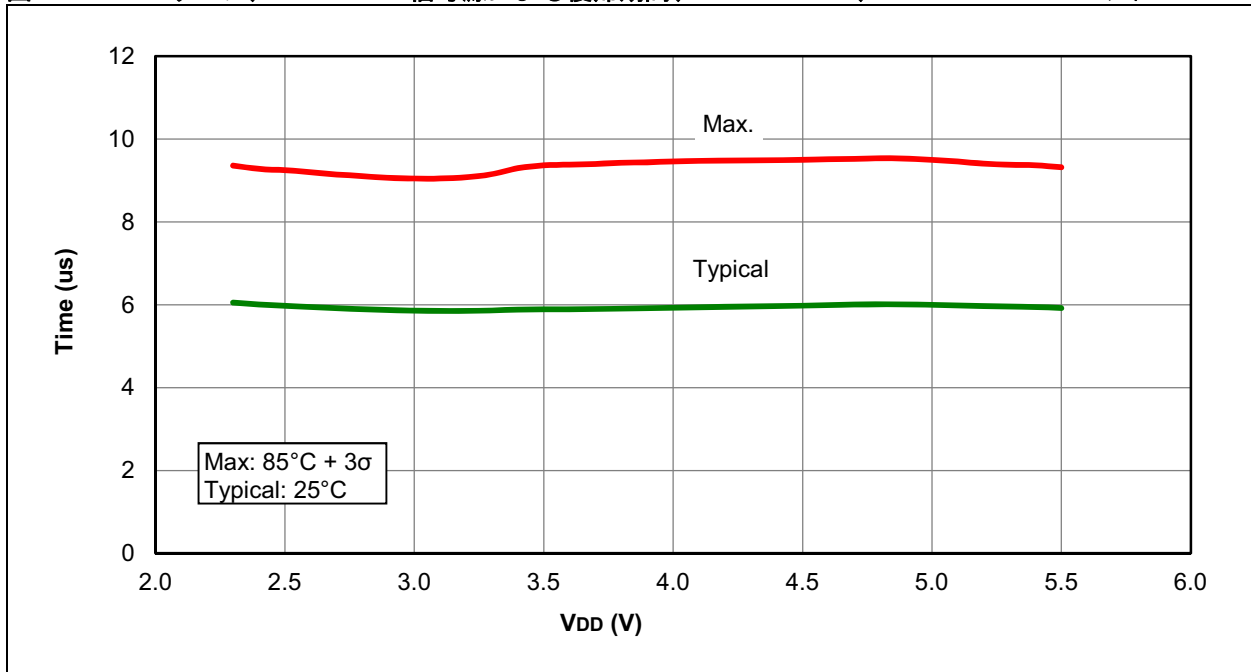


図 37-24: スリープ、HFINTOSC 信号源による復帰期間、VREGPM = 0、PIC16F1764/5/8/9 のみ



38.0 開発サポート

PIC® マイクロコントローラ (MCU) と dsPIC® デジタルシグナルコントローラ (DSC) は、以下に示す各種ソフトウェア / ハードウェア開発ツールによってサポートされています。

- Integrated Development Environment(統合開発環境)
 - MPLAB® X IDE ソフトウェア
- コンパイラ / アセンブラ / リンカ
 - MPLAB XC コンパイラ
 - MPASM™ アセンブラ
 - MPLINK™ オブジェクト リンカ / MPLIB™ オブジェクト ライブラリアン
 - 各種デバイスファミリ用 MPLAB アセンブラ / リンカ / ライブラリアン
- シミュレータ
 - MPLAB X SIM ソフトウェア シミュレータ
- エミュレータ
 - MPLAB REAL ICE™ インサーキット エミュレータ
- インサーキット デバッグ / プログラマ
 - MPLAB ICD 3
 - PICKit™ 3
- デバイス プログラマ
 - MPLAB PM3 デバイス プログラマ
- 低コストのデモボード、開発ボード、評価キット、スタータキット
- サードパーティ製開発ツール

38.1 MPLAB X 統合開発環境ソフトウェア

MPLAB X IDE は、Microchip 社およびサードパーティのソフトウェア / ハードウェア開発ツールに対する共通のグラフィカル ユーザ インターフェイスで、Windows®、Linux、Mac OS® Xで動作します。NetBeans IDE をベースにした MPLAB X IDE は、無償のソフトウェア コンポーネントとプラグインを豊富に揃えた全く新しい IDE で、高性能アプリケーションの開発とデバッグを実行できます。ツール間の移行も、ソフトウェア シミュレータからハードウェア デバッグ / プログラミング ツールへのアップグレードも、このシームレスなユーザ インターフェイスで簡単に行う事ができます。

プロジェクト管理、視覚的なコールグラフ、設定可能なウオッチ ウィンドウ、多機能エディタ (コード補完、コンテキスト メニュー等) を完備した MPLAB X IDE は非常に柔軟性が高く、初めてのユーザでも簡単に使えます。複数のプロジェクトを複数のツールで開きながら同時にデバッグを行う機能も備える等、MPLAB X IDE はベテランユーザのニーズにも対応します。

多機能エディタ :

- 構文の色分け表示
- 入力中に候補とヒントを表示するスマートコード補完
- ユーザ定義のルールに基づくコードの自動フォーマット処理
- リアルタイムの構文解析

カスタマイズ可能で使いやすいインターフェイス :

- ツールバー、ツールバーのボタン、ウィンドウ、ウィンドウ配置等のインターフェイスを完全にカスタマイズ可能
- コールグラフ ウィンドウ

プロジェクト ベースのワークスペース :

- 複数プロジェクト
- 複数ツール
- 複数設定
- 同時デバッグ セッション

ファイル履歴とバグトラッキング :

- ローカルファイル履歴機能
- Bugzilla(バグ追跡システム) をサポート

38.2 MPLAB XC コンパイラ

MPLAB XC コンパイラは、Microchip 社の 8/16/32 ビット MCU および DSC デバイス全てに対応する完全な ANSI C コンパイラです。これらのコンパイラは強力な統合機能と優れたコード最適化機能を備え、容易に使えます。MPLAB XC コンパイラは Windows、Linux、Mac OS X で動作します。

また、MPLAB X IDE 用に最適化されたデバッグ情報を出力できるため、ソースレベルのデバッグも容易です。

この MPLAB XC コンパイラの Free エディションは、全てのデバイスとコマンドをサポートし、使用期間とメモリ量に制約はありません。また、ほとんどのアプリケーションに十分なコード最適化機能も備えています。

MPLAB XC コンパイラは、アセンブラ、リンカ、ユーティリティを含みます。このアセンブラが生成した再配置可能なオブジェクト ファイルをアーカイブまたは他の再配置可能なオブジェクト ファイルとリンクして、実行ファイルを生成します。MPLAB XC コンパイラはこのアセンブラを使ってオブジェクト ファイルを生成します。このアセンブラには、主に以下の特長があります。

- デバイスの全命令セットのサポート
- 固定 / 浮動小数点データのサポート
- コマンドラインインターフェイス
- 豊富なディレクティブセット
- 柔軟なマクロ言語
- MPLAB X IDE との互換性

38.3 MPASM アセンブラ

MPASM アセンブラは、PIC10/12/16/18 MCU 用のフル機能汎用マクロアセンブラです。

MPASM アセンブラは、MPLINK オブジェクト リンカ用の再配置可能なオブジェクト ファイル、Intel® 標準 HEX ファイル、メモリ使用状況とシンボル参照を詳述する MAP ファイル、ソース行と生成後のマシンコードを含む絶対 LST ファイル、デバッグ用の COFF ファイルを生成します。

MPASM アセンブラには、以下の特長があります。

- MPLAB X IDE プロジェクトへの統合
- ユーザ定義マクロによるアセンブリコードの最適化アセンブリコード
- 多用途ソースファイルに対応する条件付きアセンブリ
- アセンブリ プロセスを完全に制御できるディレクティブ

38.4 MPLINK オブジェクト リンカ / MPLIB オブジェクト ライブラリアン

MPLINK オブジェクト リンカは、MPASM アセンブラで生成された再配置可能なオブジェクトを結合します。このオブジェクト リンカは、リンカスクリプトのディレクティブを使って、プリコンパイル済みのライブラリ内の再配置可能なオブジェクトをリンクできます。

MPLIB オブジェクト ライブラリアンは、プリコンパイル済みコードのライブラリ ファイルの作成と変更を管理します。ライブラリのルーチンをソースファイルから呼び出すと、そのルーチンが含まれているモジュールのみがアプリケーションとリンクされます。これにより、大きなライブラリを各種アプリケーションで効率的に使えます。

オブジェクト リンカ / ライブラリには、以下の特長があります。

- 多数の小さなファイルをリンクするのではなく、1 つのライブラリを効率的にリンクする
- 関連するモジュールをグループ化する事により、コードの保守性が向上する
- モジュールのリスト作成、置換、削除、抽出が容易なライブラリを柔軟に作成する

38.5 各種デバイスファミリ用 MPLAB アセンブラ / リンカ / ライブラリアン

MPLAB アセンブラは、PIC24、PIC32、dsPIC DSC 用のシンボリック アセンブリ言語から、再配置可能なマシンコードを作成します。MPLAB XC コンパイラはこのアセンブラを使ってオブジェクト ファイルを生成します。このアセンブラが生成した再配置可能なオブジェクト ファイルをアーカイブまたは他の再配置可能なオブジェクト ファイルとリンクして、実行ファイルを生成します。このアセンブラには、主に以下の特長があります。

- デバイスの全命令セットのサポート
- 固定 / 浮動小数点データのサポート
- コマンドラインインターフェイス
- 豊富なディレクティブセット
- 柔軟なマクロ言語
- MPLAB X IDE との互換性

38.6 MPLAB X SIM ソフトウェア シミュレータ

MPLAB X SIM ソフトウェア シミュレータには、PIC MCU と dsPIC DSC を命令レベルでシミュレートする機能があり、PC 環境でコードを開発できます。任意の命令に対してデータ領域を検証または変更でき、総合的なスティミュラス コントローラから外部信号を印加できます。レジスタをファイルに記録して、より詳細な実行時解析が可能です。また、トレースバッファとロジック アナライザ ディスプレイを使って、プログラムの実行、I/O アクティビティ、ほとんどの周辺モジュールと内部レジスタの記録と確認ができるため、シミュレータとしての完成度がより向上しています。

MPLAB X SIM ソフトウェア シミュレータは、MPLAB XC コンパイラ、MPASM/MPLAB アセンブラを使ったシンボリック デバッグを完全サポートしています。このソフトウェア シミュレータは、ハードウェアラボ環境外での柔軟なコード開発とデバッグを可能にする経済的で優れたソフトウェア開発ツールです。

38.7 MPLAB REAL ICE インサーキット エミュレータ システム

MPLAB REAL ICE インサーキット エミュレータ システムは、Microchip 社のフラッシュ DSC と MCU 用に Microchip 社が提供する次世代高速エミュレータです。このエミュレータでは、MPLAB X IDE の強力な使いやすいグラフィカル ユーザ インターフェイスを利用して、8/16/32 ビット MCU と DSC の全てをデバッグ/プログラムできます。

このエミュレータをハイスピード USB 2.0 インターフェイスで設計エンジニアの PC に接続し、ターゲット デバイスとはインサーキット デバッグシステムと互換の RJ-11 コネクタか、高速で耐ノイズ性に優れた最新の LVDS インターフェイス (CAT5) で接続します。

このエミュレータの更新用ファームウェアは、MPLAB X IDE からダウンロードできます。MPLAB REAL ICE は、高速エミュレーション、実行時変数ウォッチ、トレース解析、複雑なブレークポイント、論理プローブ、高耐久性のプローブ インターフェイス、接続ケーブルの長尺対応 (最長 3 m) 等、競合他社のエミュレータに比べて多くの利点があります。

38.8 MPLAB ICD 3 インサーキット デバッグシステム

MPLAB ICD 3 インサーキット デバッグシステムは、Microchip 社のフラッシュ DSC と MCU に対応した、非常に対費用効果の高い高速ハードウェア デバッグ / プログラムです。このデバッグでは、MPLAB IDE の強力な使いやすいグラフィカル ユーザ インターフェイスを利用して、PIC フラッシュ マイクロコントローラと dsPIC DSC をデバッグ / プログラムできます。

MPLAB ICD 3 インサーキット デバッグのプロープは、設計エンジニアの PC との接続にハイスピード USB 2.0 インターフェイスを使い、ターゲット デバイスとの接続には MPLAB ICD 2/MPLAB REAL ICE システムと互換のコネクタ (RJ-11) を使います。MPLAB ICD 3 は全ての MPLAB ICD 3 ヘッダをサポートしています。

38.9 PICKit 3 インサーキット デバッグ / プログラム

MPLAB PICKit 3 は、MPLAB IDE の強力なグラフィカル ユーザ インターフェイスを利用して PIC および dsPIC フラッシュ マイクロコントローラをデバッグ / プログラムできる低価格なツールです。MPLAB PICKit 3 と設計エンジニアの PC の接続にはフルスピード USB インターフェイスを使います。ターゲット デバイスとの接続には、MPLAB ICD 3/MPLAB REAL ICE と互換の Microchip 社のデバッグコネクタ (RJ-11) を使います。このコネクタは 2 本のデバイス I/O ピンとリセットラインを使って、インサーキット デバッグとインサーキット シリアル プログラミング™ (ICSP™) を実現します。

38.10 MPLAB PM3 デバイス プログラム

MPLAB PM3 デバイス プログラムは CE 準拠のユニバーサル デバイス プログラムで、VDDMIN と VDDMAX でのプログラマブル電圧検証によって最大限の信頼性を確保します。このデバイス プログラムは、メニューとエラーメッセージを表示する大型 LCD (128 x 64) と、各種パッケージタイプに対応するための脱着可能なモジュール式ソケット アセンブリを備えます。ICSP ケーブル アセンブリは標準で付属しています。スタンドアロンモードでは、MPLAB PM3 デバイス プログラムを PC に接続せずに、PIC MCU の読み出し、検証、プログラムが可能です。このモードでコード保護も設定できます。MPLAB PM3 とホスト PC との接続には、RS-232 または USB ケーブルを使います。さらに、大容量メモリデバイスの高速プログラミングを可能にする高速通信と最適化されたアルゴリズムを備え、ファイル保存とデータ アプリケーションのための MMC カードを内蔵しています。

38.11 デモボード、開発ボード、 評価キット、スタータキット

各種 PIC MCU と dsPIC DSC に対応する、豊富なデモボード、開発ボード、評価用ボードによって、完全に機能するシステムでアプリケーションを迅速に開発できます。ほとんどのボードは、カスタム回路を追加するためのプロトタイプ領域を備えています。また、付属のアプリケーション ファームウェアとソースコードを使って動作を評価できます。これらを編集して使う事もできます。

これらのボードは LED、温度センサ、スイッチ、スピーカ、RS-232 インターフェイス、LCD、ポテンショメータ、増設 EEPROM メモリ等の各種機能をサポートします。

デモボードと開発ボードは、カスタム回路の試作と各種マイクロコントローラ アプリケーションの学習教材として使う事ができます。

PICDEM™ と dsPICDEM™ デモ/開発ボードシリーズの回路の他に、Microchip 社ではアナログフィルタ設計、KEELOQ® セキュリティ IC、CAN、IrDA®、PowerSmart バッテリ管理、SEEVAL® 評価システム、 $\Delta\Sigma$ 型 ADC、流量感知等、各種アプリケーションに対応する評価キットとデモソフトウェアを取り揃えています。

また、特定のデバイスを体験するために必要なもの全てを収めたスタータキットも提供しています。スタータキットは通常、1つのアプリケーションとデバッグ機能の全てを1つのボードに搭載した形で提供します。

デモボード、開発ボード、評価キットの全一覧は、Microchip 社のウェブページサイト (www.microchip.com) をご覧ください。

38.12 サードパーティ製開発ツール

Microchip 社は、サードパーティ製のツールも数多く提供しています。これらのツールを使うと、他では得られない便利な機能が得られます。

- デバイス プログラマ/ギャング プログラマ: SoftLog 社、CCS 社等の製品
- ソフトウェア ツール: Gimpel 社、Trace Systems 社等の製品
- プロトコル アナライザ: Saleae 社、Total Phase 社等の製品
- デモボード: MikroElektronika 社、Digilent® 社、Olimex 社等の製品
- 組み込み Ethernet ソリューション: EZ Web Lynx 社、WIZnet 社、IPLogika® 社等の製品

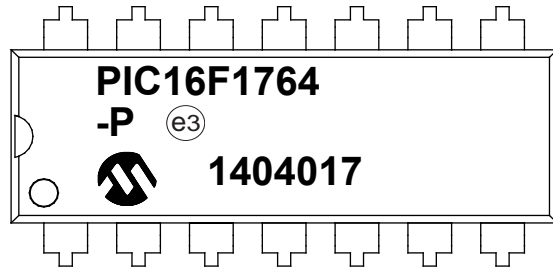
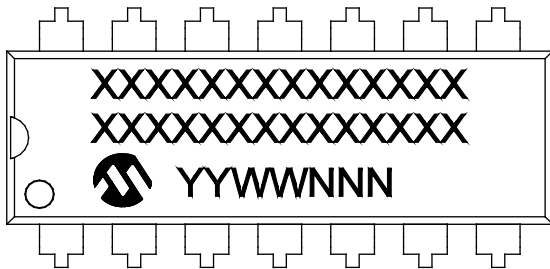
PIC16(L)F1764/5/8/9

39.0 パッケージ情報

39.1 パッケージマーキング情報

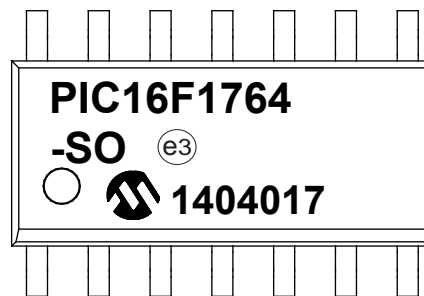
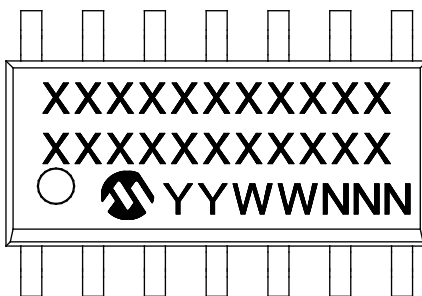
14 ピン PDIP (300 mil)

例



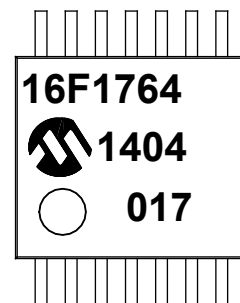
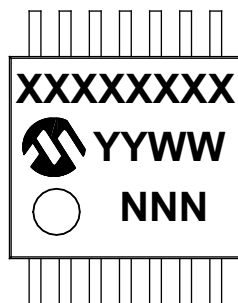
14 ピン SOIC (3.90 mm)

例



14 ピン TSSOP (4.4 mm)

例



凡例: XX...X お客様固有情報
 Y 年コード (西暦の下1桁)
 YY 年コード (西暦の下2桁)
 WW 週コード (1月の第1週を「01」とする)
 NNN 英数字のトレーサビリティコード
 * (e3) つや消し錫 (Sn) の使用を示す鉛フリーの JEDEC® マーク
 このパッケージは鉛フリーです。鉛フリー JEDEC® マーク (e3) は外箱に表記しています。

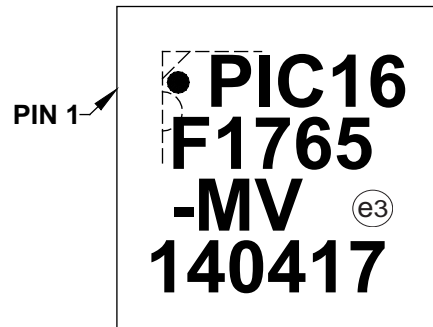
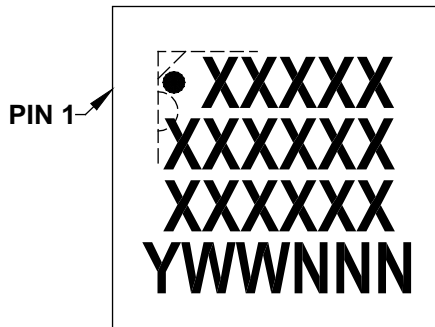
Note: Microchip 社の製品番号が 1 行に収まりきらない場合は複数行を使います。
 この場合、お客様固有情報に使える文字数が制限されます。

PIC16(L)F1764/5/8/9

パッケージ マーキング情報 (続き)

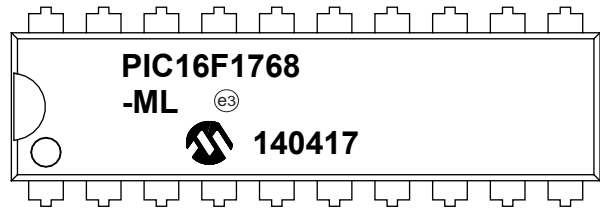
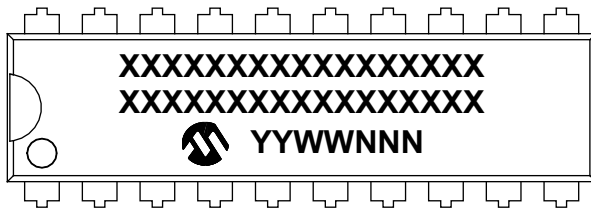
16 ピン QFN (4x4x0.9 mm)

例



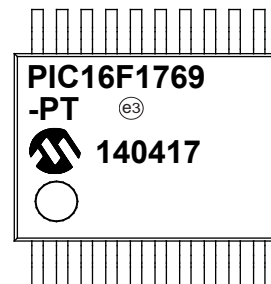
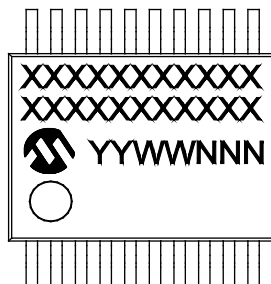
20 ピン PDIP (300 mil)

例



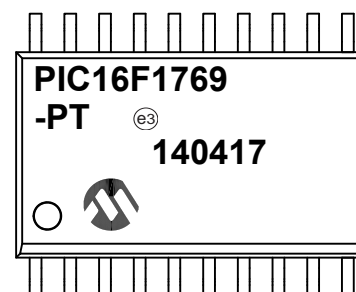
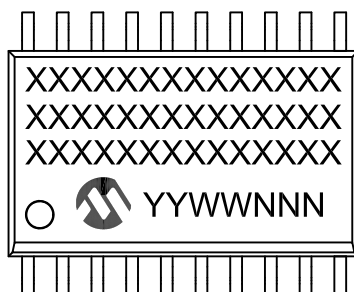
20 ピン SSOP (5.30 mm)

例



20 ピン SOIC (7.50 mm)

例

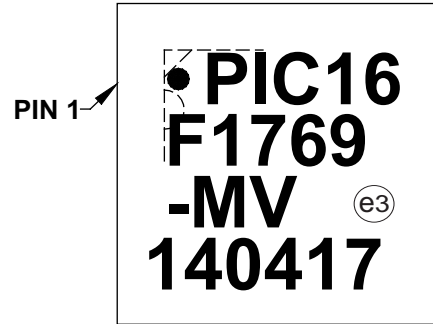
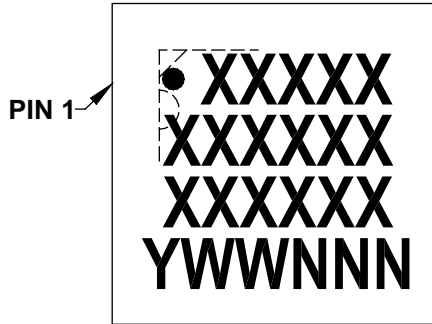


PIC16(L)F1764/5/8/9

パッケージ マーキング情報 (続き)

20 ピン QFN (4x4x0.9 mm)

例



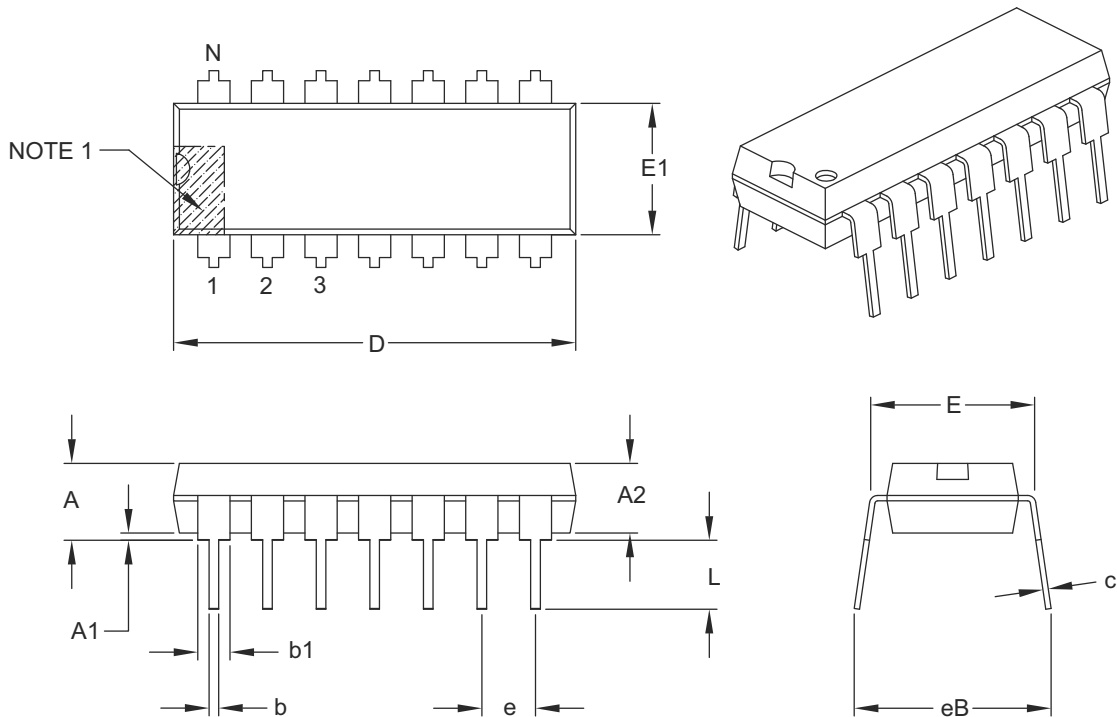
PIC16(L)F1764/5/8/9

39.2 パッケージの詳細

以下に各パッケージの技術的詳細を記載します。

14 ピン プラスチック デュアル インライン (P) - 300 mil ボディ [PDIP]

Note: 最新のパッケージ図面については、以下のウェブページにある『Microchip Packaging Specification』を参照してください。
<http://www.microchip.com/packaging>



寸法限界	単位	インチ		
		最小	公称	最大
ピン数	N	14		
ピッチ	e	.100 BSC		
トップからシーティングまで	A	-	-	.210
モールド パッケージ厚	A2	.115	.130	.195
ベースからシーティング プレーンまで	A1	.015	-	-
ショルダ間幅	E	.290	.310	.325
モールド パッケージ幅	E1	.240	.250	.280
全長	D	.735	.750	.775
先端からシーティング プレーンまで	L	.115	.130	.150
ピン厚	c	.008	.010	.015
上側ピン幅	b1	.045	.060	.070
下側ピン幅	b	.014	.018	.022
全幅 §	eB	-	-	.430

Notes:

1. ピン 1 のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
2. § 重要な特性です。
3. D と E1 の寸法はバリを含みません。モールド フラッシュまたは突出部は各側で 0.010" (0.254 mm) 以下とします。
4. 寸法と許容誤差は ASME Y14.5M に準拠しています。

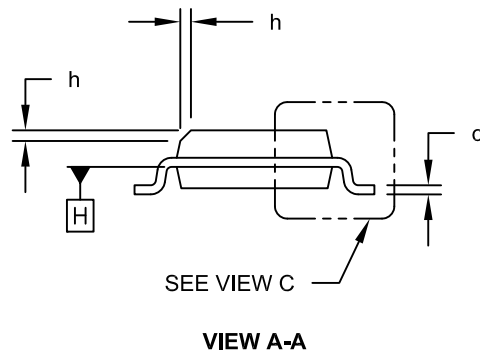
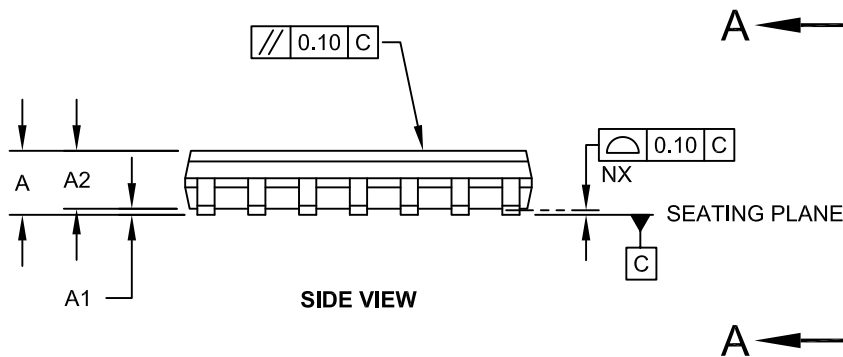
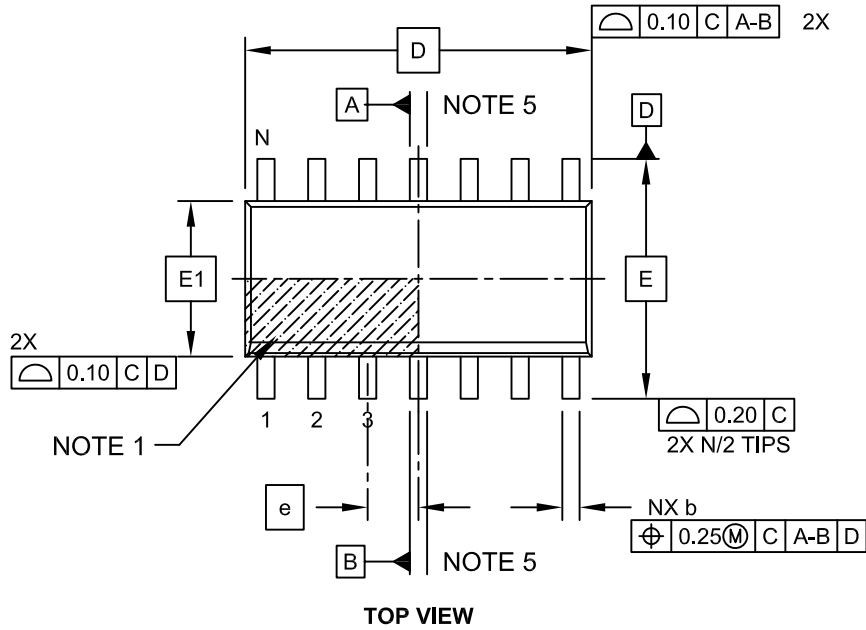
BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing C04-005B

PIC16(L)F1764/5/8/9

14 ピン プラスチック スモール アウトライン (SL) - ナロー、3.90 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>

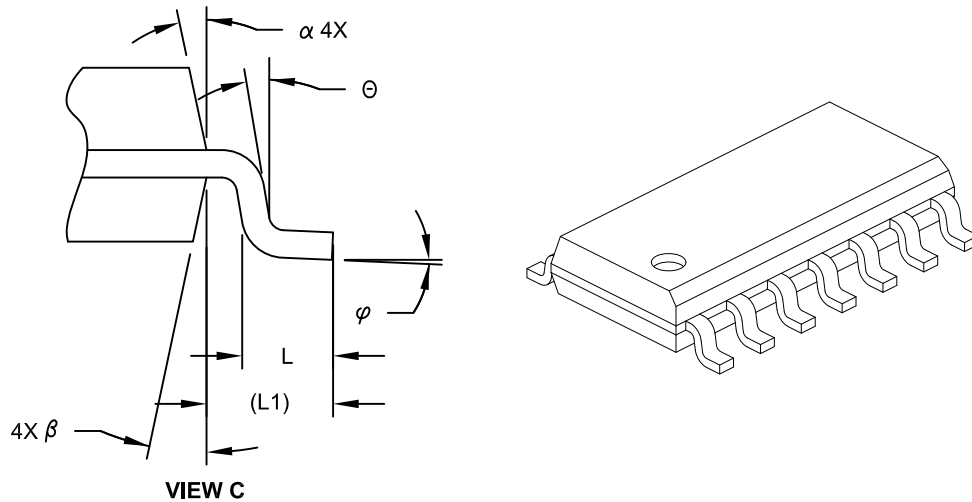


Microchip Technology Drawing No.C04-065C シート 1/2

PIC16(L)F1764/5/8/9

14 ピン プラスチック スモール アウトライン (SL) - ナロー、3.90 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packageing>



寸法	単位	ミリメートル		
		最小	公称	最大
ピン数	N	14		
ピッチ	e	1.27 BSC		
全高	A	-	-	1.75
モールドパッケージ厚	A2	1.25	-	-
スタンドオフ §	A1	0.10	-	0.25
全幅	E	6.00 BSC		
モールドパッケージ幅	E1	3.90 BSC		
全長	D	8.65 BSC		
面取り部 (オプション)	h	0.25	-	0.50
足長	L	0.40	-	1.27
フットプリント	L1	1.04 REF		
ピン角	θ	0°	-	-
足角	φ	0°	-	8°
ピン厚	c	0.10	-	0.25
ピン幅	b	0.31	-	0.51
モールドドラフト角トップ	α	5°	-	15°
モールドドラフト角ボトム	β	5°	-	15°

Notes:

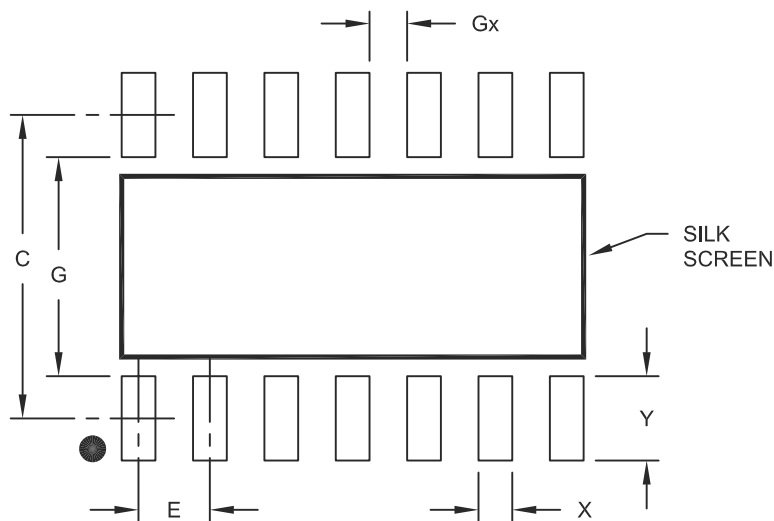
- ピン 1 のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- § 重要な特性です。
- D の寸法はバリを含みません。これらは各側で 0.15 mm 以下とします。E1 の寸法はバリを含みません。これらは各側で 0.25 mm 以下とします。
- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示
 REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値
- データ A と B はデータ H の位置で決まります。

Microchip Technology Drawing No.C04-065C シート 2/2

PIC16(L)F1764/5/8/9

14 ピン プラスチック スモール アウトライン (SL) - ナロー、3.90 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

	単位	ミリメートル		
		寸法限界	最小	公称
コンタクトピッチ	E	1.27 BSC		
コンタクトパッド間隔	C		5.40	
コンタクトパッド幅	X			0.60
コンタクトパッド長	Y			1.50
パッド間距離	Gx	0.67		
パッド間距離	G	3.90		

Notes:

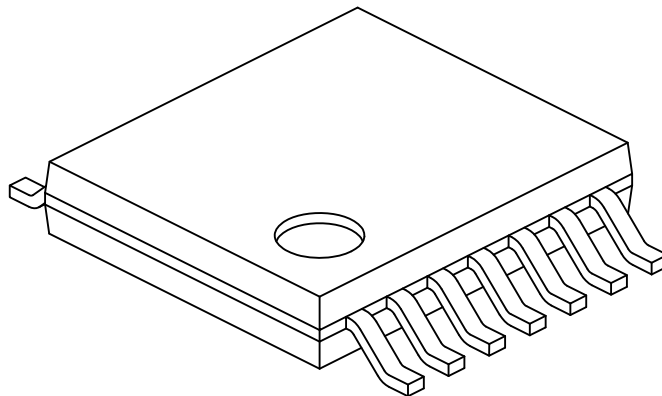
1. 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing No.C04-2065A

PIC16(L)F1764/5/8/9

14ピンプラスチック薄型シュリンク スモール アウトライン (ST) - 4.4 mm ボディ [TSSOP]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



	単位	ミリメートル		
		寸法限界	最小	公称
ピン数	N	14		
ピッチ	e	0.65 BSC		
全高	A	-	-	1.20
モールドパッケージ厚	A2	0.80	1.00	1.05
スタンドオフ	A1	0.05	-	0.15
全幅	E	6.40 BSC		
モールドパッケージ幅	E1	4.30	4.40	4.50
モールドパッケージ長	D	4.90	5.00	5.10
足長	L	0.45	0.60	0.75
フットプリント	(L1)	1.00 REF		
足角	ϕ	0°	-	8°
ピン厚	c	0.09	-	0.20
ピン幅	b	0.19	-	0.30

Notes:

- ピン1のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- DとE1の寸法はバリを含みません。バリは側面から0.15 mmを超えません。
- 寸法と許容誤差はASME Y14.5Mに準拠しています。

BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

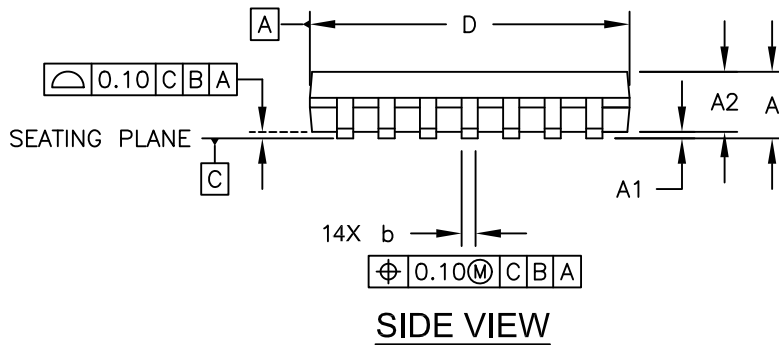
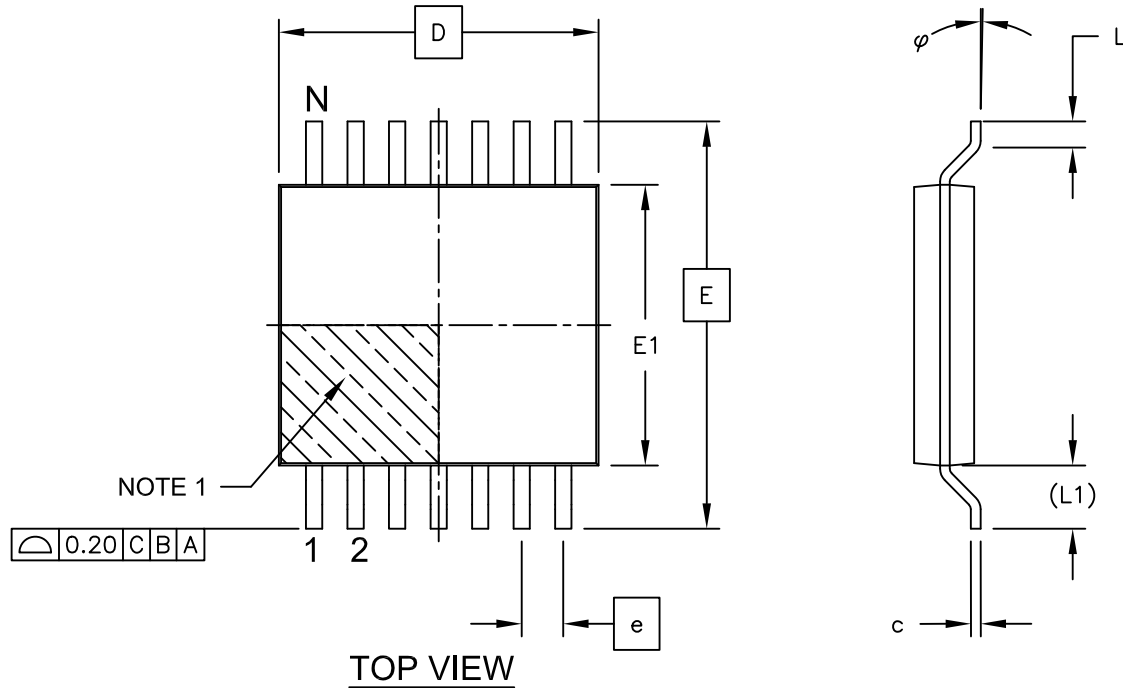
REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値

Microchip Technology Drawing No.C04-087C シート 2/2

PIC16(L)F1764/5/8/9

14ピン プラスチック薄型シュリンク スモール アウトライン (ST) - 4.4 mm ボディ [TSSOP]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>

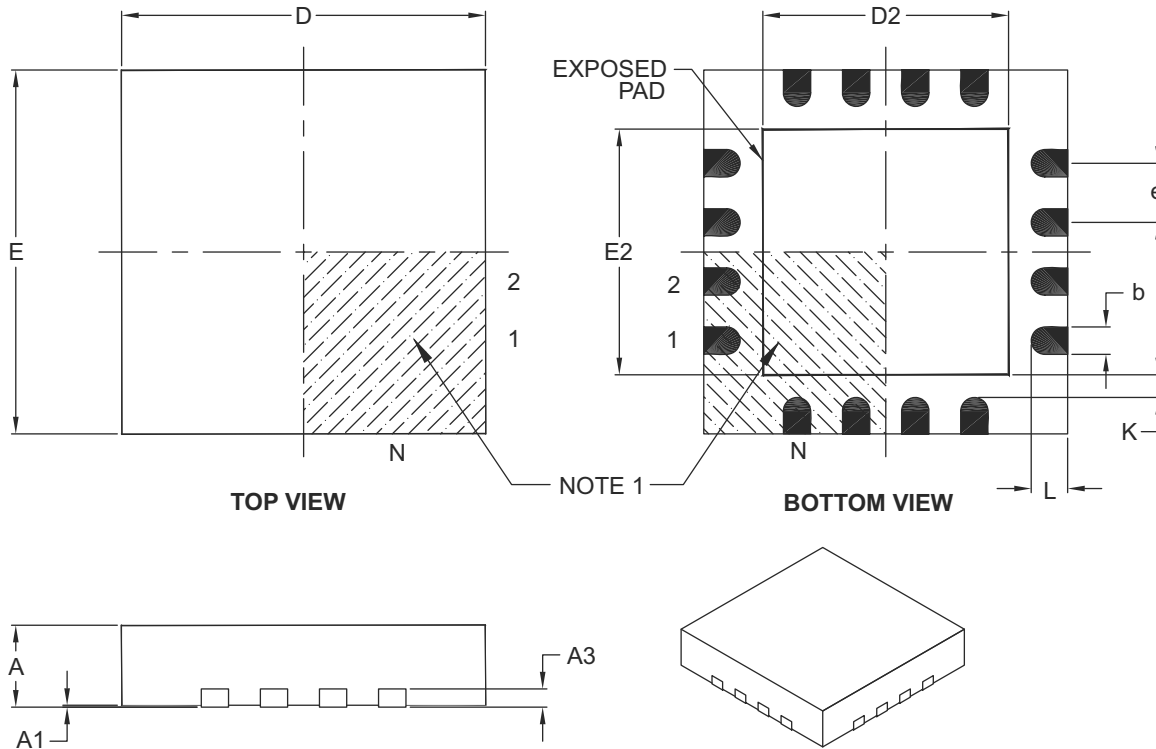


Microchip Technology Drawing C04-087C シート 1/2

PIC16(L)F1764/5/8/9

16ピンプラスチッククワッドフラット、リードレスパッケージ (ML) - 4x4x0.9 mm ボディ [QFN]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



寸法	単位	ミリメートル		
		最小	公称	最大
ピン数	N	16		
ピッチ	e	0.65 BSC		
全高	A	0.80	0.90	1.00
スタンドオフ	A1	0.00	0.02	0.05
コンタクト厚	A3	0.20 REF		
全幅	E	4.00 BSC		
露出パッド幅	E2	2.50	2.65	2.80
全長	D	4.00 BSC		
露出パッド長	D2	2.50	2.65	2.80
コンタクト幅	b	0.25	0.30	0.35
コンタクト長	L	0.30	0.40	0.50
コンタクト - 露出パッド間距離	K	0.20	-	-

Notes:

- ピン1のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- パッケージはダイサーで個片化されています。
- 寸法と許容誤差は ASME Y14.5M に準拠しています。

BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

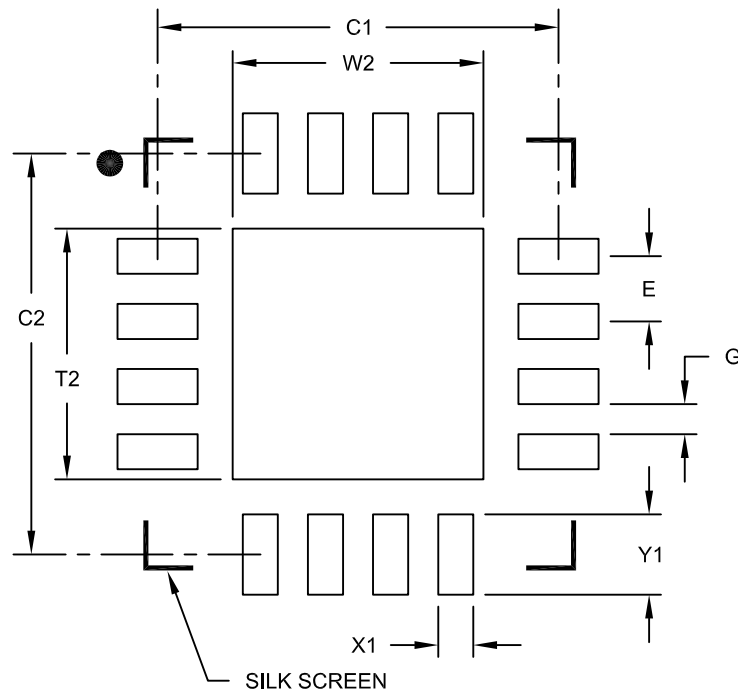
REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-127B

PIC16(L)F1764/5/8/9

16 ピン プラスチック クワッド フラット、リードレス パッケージ (ML) - 4x4x0.9 mm ボディ [QFN]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.65 BSC		
オプションのセンターパッド幅	W2			2.50
オプションのセンターパッド長	T2			2.50
コンタクトパッド間隔	C1		4.00	
コンタクトパッド間隔	C2		4.00	
コンタクトパッド幅 (X16)	X1			0.35
コンタクトパッド長 (X16)	Y1			0.80
パッド間距離	G	0.30		

Notes:

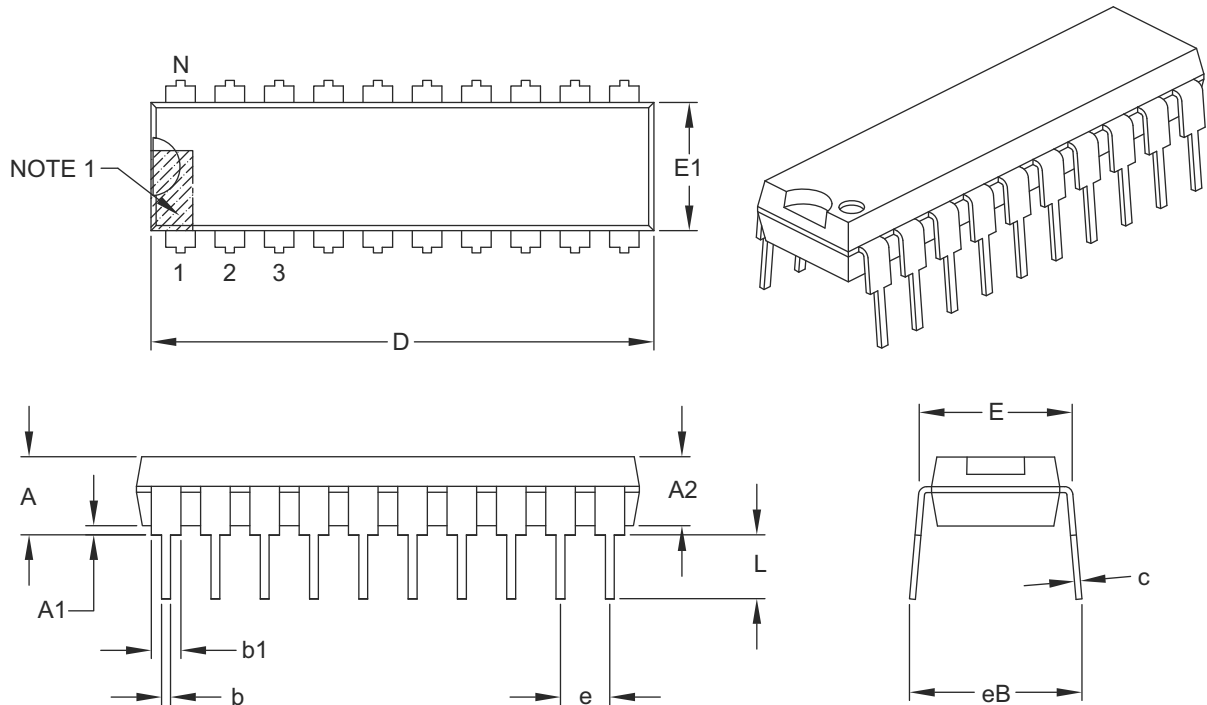
- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing No.C04-2127A

PIC16(L)F1764/5/8/9

20 ピン プラスチック デュアル インライン (P) - 300 mil ボディ [PDIP]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



寸法限界	単位	インチ		
		最小	公称	最大
ピン数	N	20		
ピッチ	e	.100 BSC		
トップからシーティング プレーンまで	A	-	-	.210
モールド パッケージ厚	A2	.115	.130	.195
ベースからシーティング プレーンまで	A1	.015	-	-
ショルダ間幅	E	.300	.310	.325
モールド パッケージ幅	E1	.240	.250	.280
全長	D	.980	1.030	1.060
先端からシーティングまで	L	.115	.130	.150
ピン厚	c	.008	.010	.015
上側ピン幅	b1	.045	.060	.070
下側ピン幅	b	.014	.018	.022
全幅 §	eB	-	-	.430

Notes:

1. ピン 1 のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
2. § 重要な特性です。
3. D と E1 の寸法はバリを含みません。モールド フラッシュまたは突出部は各側で 0.010" (0.254 mm) 以下とします。
4. 寸法と許容誤差は ASME Y14.5M に準拠しています。

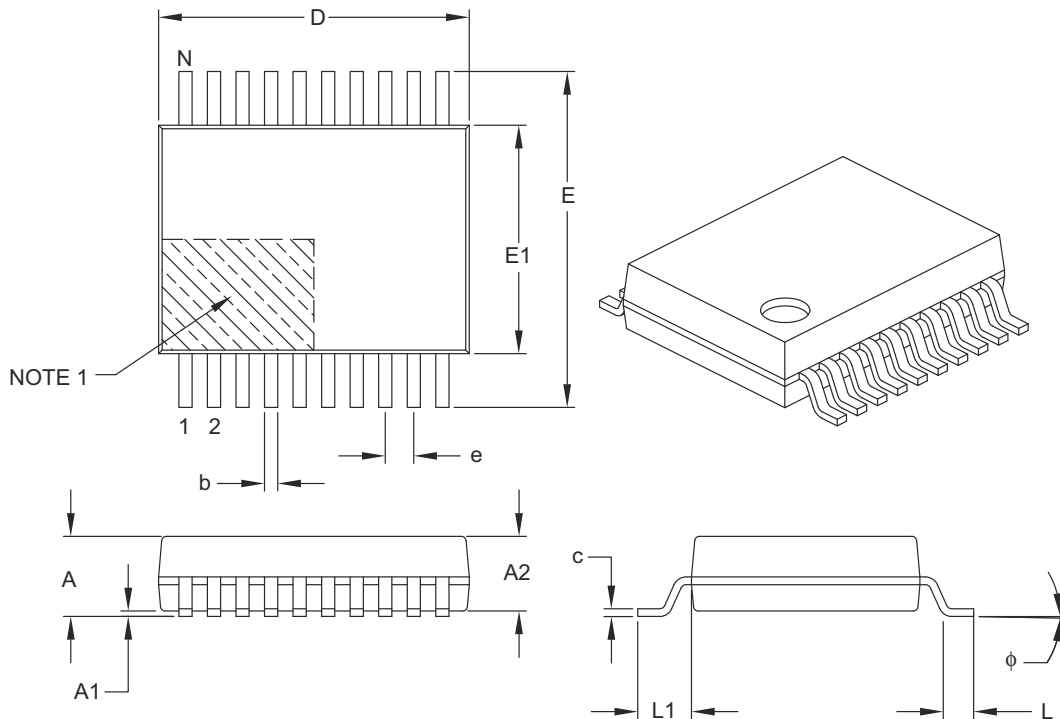
BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing C04-019B

PIC16(L)F1764/5/8/9

20ピンプラスチックシュリンクスモールアウトライン(SS) - 5.30 mm ボディ [SSOP]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
ピン数	N	20		
ピッチ	e	0.65 BSC		
全高	A	-	-	2.00
モールドパッケージ厚	A2	1.65	1.75	1.85
スタンドオフ	A1	0.05	-	-
全幅	E	7.40	7.80	8.20
モールドパッケージ幅	E1	5.00	5.30	5.60
全長	D	6.90	7.20	7.50
足長	L	0.55	0.75	0.95
フットプリント	L1	1.25 REF		
ピン厚	c	0.09	-	0.25
足角	f	0°	4°	8°
ピン幅	b	0.22	-	0.38

Notes:

- ピン1のビジュアルインデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- DとE1の寸法はバリを含みません。バリは側面から0.20 mmを超えません。
- 寸法と許容誤差はASME Y14.5Mに準拠しています。

BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

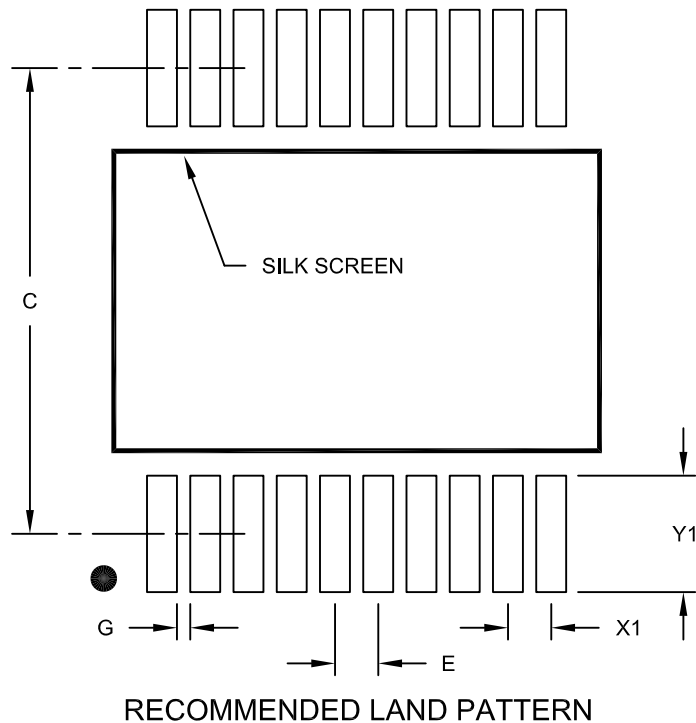
REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-072B

PIC16(L)F1764/5/8/9

20 ピン プラスチック シュリンク スモール アウトライン (SS) - 5.30 mm ボディ [SSOP]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.65 BSC		
コンタクトパッド間隔	C1		7.20	
コンタクトパッド幅 (X20)	X1			0.45
コンタクトパッド長 (X20)	Y1			1.75
パッド間距離	G	0.20		

Notes:

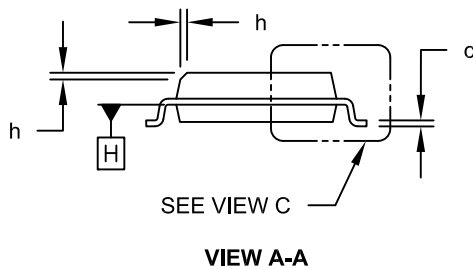
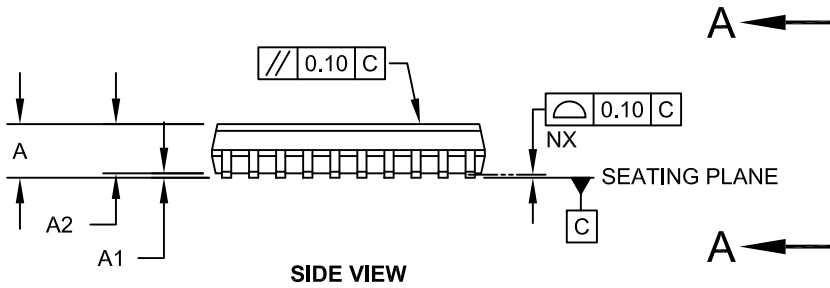
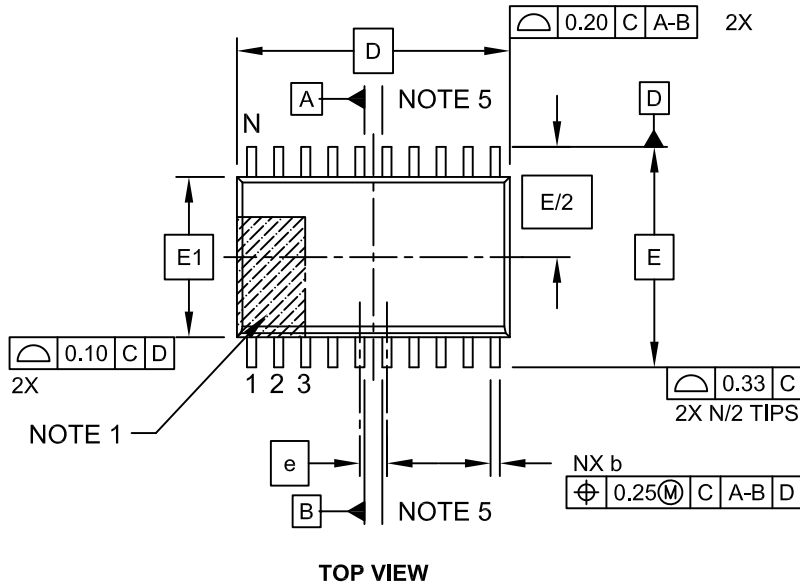
- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing No.C04-2072A

PIC16(L)F1764/5/8/9

20 ピン プラスチック スモール アウトライン (SO) - ワイド、7.50 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>

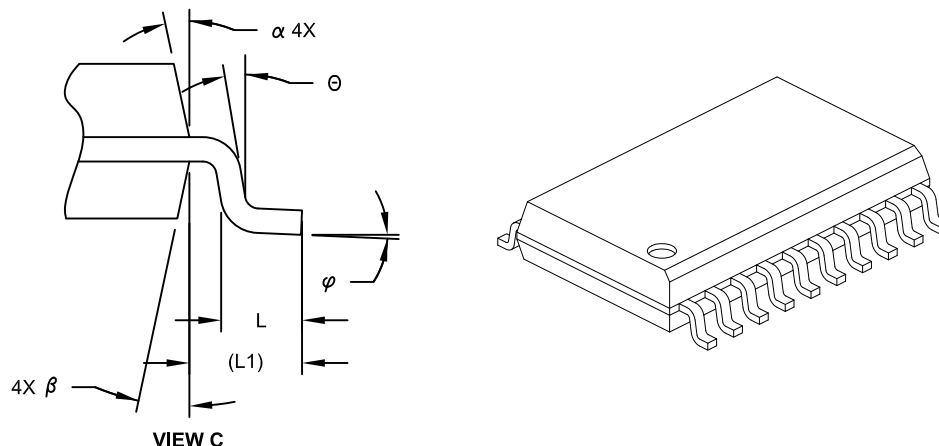


Microchip Technology Drawing C04-094C シート 1/2

PIC16(L)F1764/5/8/9

20ピンプラスチック スモール アウトライン (SO) - ワイド、7.50 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



単位 寸法限界	ミリメートル		
	最小	公称	最大
ピン数	N	20	
ピッチ	e	1.27 BSC	
全高	A	-	2.65
モールドパッケージ厚	A2	2.05	-
スタンドオフ §	A1	0.10	0.30
全幅	E	10.30 BSC	
モールドパッケージ幅	E1	7.50 BSC	
全長	D	12.80 BSC	
面取り部 (オプション)	h	0.25	0.75
足長	L	0.40	1.27
フットプリント	L1	1.40 REF	
ピン角	θ	0°	-
足角	φ	0°	8°
ピン厚	c	0.20	0.33
ピン幅	b	0.31	0.51
モールドドラフト角トップ	α	5°	15°
モールドドラフト角ボトム	β	5°	15°

Notes:

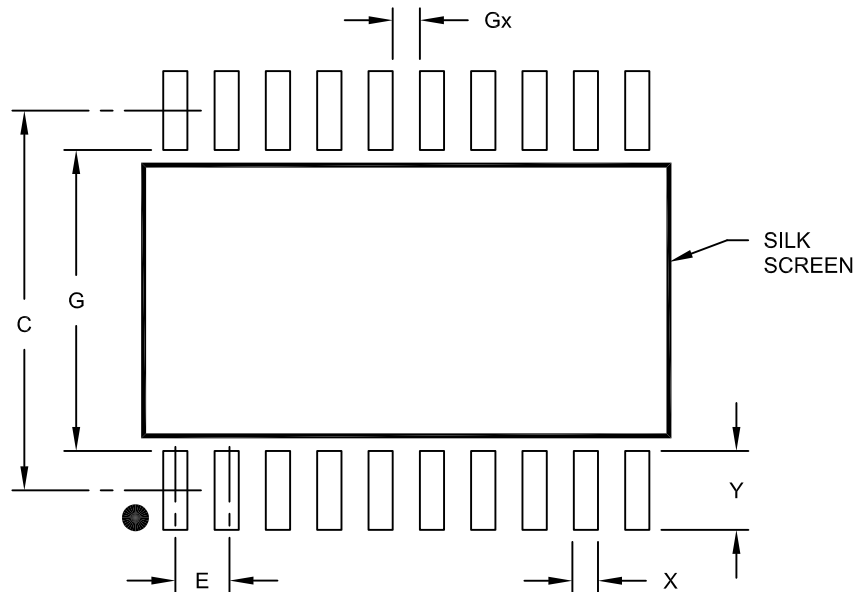
- ピン 1 のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- § 重要な特性です。
- D の寸法はバリを含みません。これらは各側で 0.15 mm 以下とします。E1 の寸法はバリを含みません。これらは各側で 0.25 mm 以下とします。
- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示
 REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値
- データ A と B はデータ H の位置で決まります。

Microchip Technology Drawing No.C04-094C シート 2/2

PIC16(L)F1764/5/8/9

20ピンプラスチック スモール アウトライン (SO) - ワイド、7.50 mm ボディ [SOIC]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

	単位	ミリメートル		
		寸法限界	最小	公称
コンタクトピッチ	E		1.27 BSC	
コンタクトパッド間隔	C		9.40	
コンタクトパッド幅 (X20)	X			0.60
コンタクトパッド長 (X20)	Y			1.95
パッド間距離	Gx	0.67		
パッド間距離	G	7.45		

Notes:

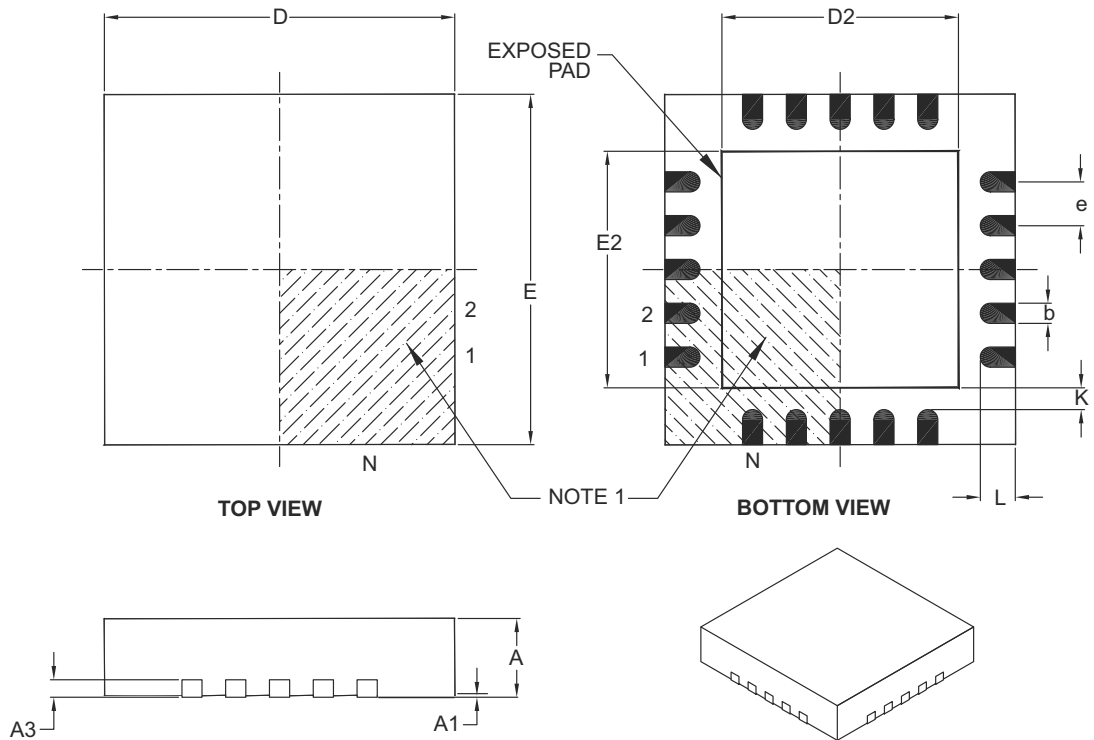
- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing No.C04-2094A

PIC16(L)F1764/5/8/9

20ピンプラスチッククワッドフラット、リードレスパッケージ (ML) - 4x4x0.9 mm ボディ [QFN]

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packageing>



	単位	ミリメートル		
		寸法限界	最小	公称
ピン数	N	20		
ピッチ	e	0.50 BSC		
全高	A	0.80	0.90	1.00
スタンドオフ	A1	0.00	0.02	0.05
コンタクト厚	A3	0.20 REF		
全幅	E	4.00 BSC		
露出パッド幅	E2	2.60	2.70	2.80
全長	D	4.00 BSC		
露出パッド長	D2	2.60	2.70	2.80
コンタクト幅	b	0.18	0.25	0.30
コンタクト長	L	0.30	0.40	0.50
コンタクト - 露出パッド間距離	K	0.20	-	-

Notes:

- ピン 1 のビジュアル インデックスの場所にはばらつきがありますが、必ず斜線部分内にあります。
- パッケージは切削切り出しされています。
- 寸法と許容誤差は ASME Y14.5M に準拠しています。

BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

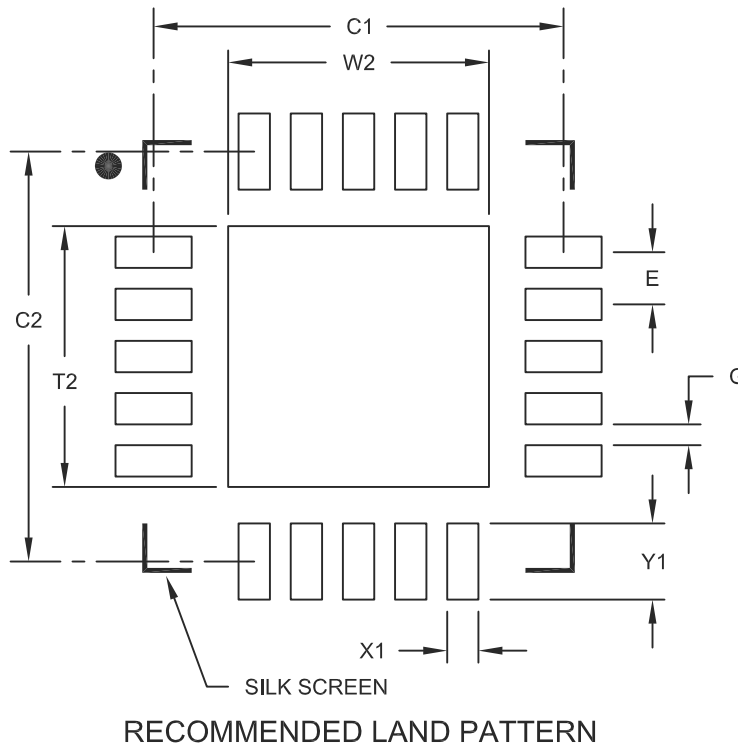
REF: 参考寸法、通常は許容誤差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-126B

PIC16(L)F1764/5/8/9

20 ピン プラスチック クワッド フラット、リードレス パッケージ (ML) - 4x4 mm ボディ [QFN]
 コンタクト長 0.40 mm

Note: 最新のパッケージ図面については、以下のウェブページにある「Microchip Packaging Specification (Microchip 社パッケージ仕様)」を参照してください。
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.50 BSC		
オプションのセンターパッド幅	W2			2.50
オプションのセンターパッド長	T2			2.50
コンタクトパッド間隔	C1		3.93	
コンタクトパッド間隔	C2		3.93	
コンタクトパッド幅	X1			0.30
コンタクトパッド長	Y1			0.73
パッド間距離	G	0.20		

Notes:

- 寸法と許容誤差は ASME Y14.5M に準拠しています。
 BSC: 基本寸法、理論的に正確な値、許容誤差なしで表示

Microchip Technology Drawing No.C04-2126A

補遺 A: データシート改訂履歴

リビジョン A (2014 年 12 月)

本書の初版です。

MICROCHIP 社ウェブサイト

Microchip 社は自社が運営するウェブサイト (www.microchip.com) を通してオンライン サポートを提供しています。このウェブサイトからファイルや情報を簡単に入手できます。一般的なインターネット ブラウザから以下の内容をご覧になれます。

- **製品サポート** - データシートとエラッタ、アプリケーション ノートとサンプル プログラム、設計リソース、ユーザガイドとハードウェア サポート文書、最新のソフトウェアと過去のソフトウェア
- **技術サポート** - よく寄せられる質問 (FAQ)、技術サポートのご依頼、オンライン ディスカッション グループ、Microchip 社コンサルタント プログラム メンバーの一覧
- **ご注文とお問い合わせ** - 製品セレクトと注文ガイド、最新プレスリリース、セミナー / イベントの一覧、お問い合わせ先 (営業所 / 販売代理店) の一覧

お客様向け変更通知サービス

Microchip 社のお客様向け変更通知サービスは、お客様が Microchip 社製品の最新情報を入手できるようにします。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

当サービスをご希望のお客様は、Microchip 社ウェブサイト (www.microchip.com) でご登録ください。

[Support] の [Customer Change Notification] をクリックして登録手順に従います。

カスタマサポート

Microchip 社製品をお使いのお客様は、以下のチャネルからサポートをご利用頂けます。

- 販売代理店
- 弊社営業所
- 技術サポート

サポートは販売代理店にお問い合わせください。弊社営業所にもご連絡頂けます。本書の末尾には各国営業所の一覧を記載しています。

技術サポートは下記のウェブページからもご利用頂けます。 <http://microchip.com/support>

製品識別システム

ご注文や製品の価格 / 納期に関しては、弊社または販売代理店までお問い合わせください。

製品番号	[X] ⁽¹⁾	-	X	/XX	XXX
デバイス	テープ & リール オプション		温度レンジ レンジ	パッケージ	パターン
<p>デバイス: PIC16F1764、PIC16LF1764、 PIC16F1765、PIC16LF1765、 PIC16F1768、PIC16LF1768、 PIC16F1769、PIC16LF1769</p>					
<p>テープ & リール オプション: 空欄 = 標準梱包 (チューブまたはトレイ) T = テープ & リール⁽¹⁾</p>					
<p>温度レンジ: I = -40 ~ +85 °C (産業用温度レンジ) E = -40 ~ +125 °C (拡張温度レンジ)</p>					
<p>パッケージ:⁽²⁾ ML = QFN P = PDIP SL = SOIC SO = SOIC SS = SSOP ST = TSSOP</p>					
<p>パターン: QTP、SQTP、その他のコード等 (あるいは空白のまま)</p>					

例:

- PIC16LF1764- I/P
産業用温度レンジ、
PDIP パッケージ
- PIC16F1769- E/SS
拡張温度レンジ、
SSOP パッケージ

- Note 1:** テープ & リールの識別情報は、カタログの製品番号説明でのみご確認頂けます。この識別情報は注文時に使うため、デバイスのパッケージには印刷されていません。テープ & リールが選択できるパッケージの在庫 / 供給状況は、最寄りの Microchip 社の営業所までお問い合わせください。
- 2:** 小型パッケージで提供している場合があります。小型パッケージは、www.microchip.com/packaging をご覧頂くか、弊社または代理店までお問い合わせください。

Microchip 社製デバイスのコード保護機能に関して以下の点にご注意ください。

- Microchip 社製品は、該当する Microchip 社データシートに記載の仕様を満たしています。
- Microchip 社では、通常の条件ならびに仕様に従って使用した場合、Microchip 社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解では、こうした手法は Microchip 社データシートにある動作仕様書以外の方法で Microchip 社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- Microchip 社は、コードの保全性に懸念を抱いているお客様と連携し、対応策に取り組んでいきます。
- Microchip 社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。Microchip 社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。Microchip 社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。Microchip 社は、本書の情報およびその使用に起因する一切の責任を否認します。生命維持装置あるいは生命安全用途に Microchip 社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、Microchip 社は擁護され、免責され、損害を受けない事に同意するものとします。暗黙的あるいは明示的を問わず、Microchip 社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

Microchip 社の名称とロゴ、Microchip ロゴ、dsPIC、FlashFlex、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、PIC³² ロゴ、rPIC、SST、SST ロゴ、SuperFlash、UNI/O は、米国およびその他の国における Microchip Technology Incorporated の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL、Embedded Control Solutions Company は、米国における Microchip Technology Incorporated の登録商標です。

Silicon Storage Technology は、他の国における Microchip Technology Inc. の登録商標です。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT ロゴ、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindī、MiWi、MPASM、MPF、MPLAB Certified ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rLAB、Select Mode、SQL、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA および Z-Scale は、米国およびその他の Microchip Technology Incorporated の商標です。

SQTP は、米国における Microchip Technology Incorporated のサービスマークです。

GestIC および ULPP は、Microchip Technology Inc. の子会社である Microchip Technology Germany II GmbH & Co. & KG 社の他の国における登録商標です。

その他本書に記載されている商標は各社に帰属します。

© 2015, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-63277-040-0

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==**

Microchip 社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州) の本部、設計部およびウェハー製造工場そしてカリフォルニア州とインドのデザインセンターが ISO/TS-16949:2009 認証を取得しています。Microchip 社の品質システム プロセス および手順は、PIC[®] MCU および dsPIC[®] DSC、KEELOQ[®] コード ホッピング デバイス、シリアル EEPROM、マイクロペリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関する Microchip 社の品質システムは ISO 9001:2000 認証を取得しています。

各国の営業所とサービス

北米

本社

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
技術サポート：
[http://www.microchip.com/
support](http://www.microchip.com/support)
URL:
www.microchip.com

アトランタ

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

オースティン、TX

Tel: 512-257-3370

ボストン

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

シカゴ

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

クリーブランド

Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

ダラス

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

デトロイト

Novi, MI
Tel: 248-848-4000

ヒューストン、TX

Tel: 281-894-5983

インディアナポリス

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

ロサンゼルス

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

ニューヨーク、NY

Tel: 631-435-6000

サンノゼ、CA

Tel: 408-735-9110

カナダ - トロント

Tel: 905-673-0699
Fax: 905-673-6509

アジア / 太平洋

アジア太平洋支社

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

オーストラリア - シドニー

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

中国 - 北京

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重慶

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州

Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港 SAR

Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青島

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 瀋陽

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武漢

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦門

Tel: 86-592-2388138
Fax: 86-592-2388130

中国 - 珠海

Tel: 86-756-3210040
Fax: 86-756-3210049

アジア / 太平洋

インド - バンガロール

Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

インド - ニューデリー

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

インド - プネ

Tel: 91-20-3019-1500

日本 - 大阪

Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 - 東京

Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韓国 - 大邱

Tel: 82-53-744-4301
Fax: 82-53-744-4302

韓国 - ソウル

Tel: 82-2-554-7200
Fax: 82-2-558-5932 または
82-2-558-5934

マレーシア - クアラルンプール

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

マレーシア - ペナン

Tel: 60-4-227-8870
Fax: 60-4-227-4068

フィリピン - マニラ

Tel: 63-2-634-9065
Fax: 63-2-634-9069

シンガポール

Tel: 65-6334-8870
Fax: 65-6334-8850

台湾 - 新竹

Tel: 886-3-5778-366
Fax: 886-3-5770-955

台湾 - 高雄

Tel: 886-7-213-7830

台湾 - 台北

Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

タイ - バンコク

Tel: 66-2-694-1351
Fax: 66-2-694-1350

ヨーロッパ

オーストリア - ヴェルス

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

デンマーク - コペンハーゲン

Tel: 45-4450-2828
Fax: 45-4485-2829

フランス - パリ

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

ドイツ - デュッセルドルフ

Tel: 49-2129-3766400

ドイツ - ミュンヘン

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

ドイツ - プフォルトツハイム

Tel: 49-7231-424750

イタリア - ミラノ

Tel: 39-0331-742611
Fax: 39-0331-466781

イタリア - ヴェニス

Tel: 39-049-7625286

オランダ - ドリユネン

Tel: 31-416-690399
Fax: 31-416-690340

ポーランド - ワルシャワ

Tel: 48-22-3325737

スペイン - マドリッド

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

スウェーデン - ストックホルム

Tel: 46-8-5090-4654

イギリス - ウォーキンガム

Tel: 44-118-921-5800
Fax: 44-118-921-5820